Graphs, Simplicial Complexes and Beyond: Topological Tools for Multi-agent Coordination

A Thesis
Presented to
The Academic Faculty

by

Abubakr Muhammad

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer
Engineering
Georgia Institute of Technology
December 2005

Graphs, Simplicial Complexes and Beyond: Topological Tools for Multi-agent Coordination

Approved by:

Dr Magnus Egerstedt, Adviser
School of Electrical & Computer Engineering
Georgia Institute of Technology

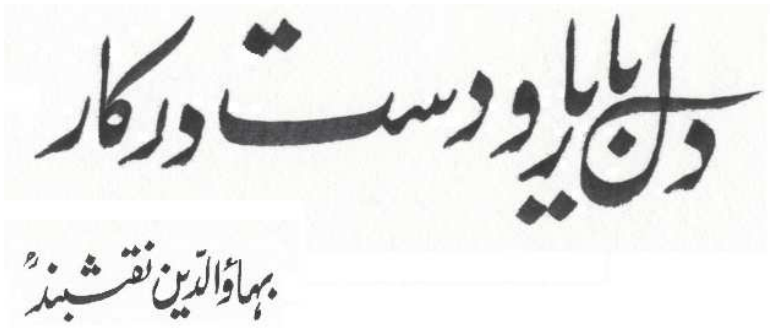
Dr Erik Verriest
School of Electrical & Computer Engineering
Georgia Institute of Technology

Dr Margaret Symington
School of Mathematics
Georgia Institute of Technology

Dr Allen Tannenbaum
School of Electrical & Computer Engineering
Georgia Institute of Technology

Dr Ayanna Howard
School of Electrical & Computer Engineering
Georgia Institute of Technology

Date Approved: Nov 10, 2005



Persian: **Dil ba Yar-o Dast dar Kar**

Translation: "The heart is with the beloved; while the hands are at work."

Bahauddin Naqshband (1318-1389 C.E.)

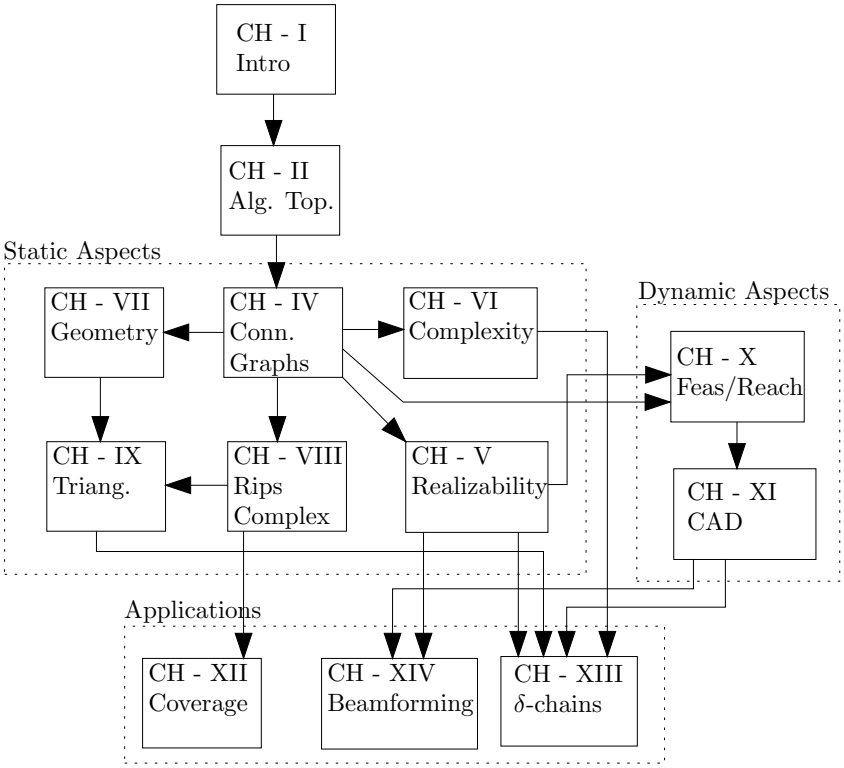
PREFACE

The ideas developed in this thesis have originated from several distinct events in my student life at Georgia Tech. These events include some thought provoking discussions, frequent travels and collaborations with some very intelligent researchers. The bulk of this work concerns the study of decentralized coordination schemes in large networks of mobile agents, using local interactions. I decided to invest my efforts into this topic after carefully realizing the potential of its applications and the recognition that there has been a lack of a solid mathematical foundation for studying such coordination problems. During the course of this work, several other researchers have focused their attention on this area, and have produced some elegant results. This thesis, however, differs from almost all other works in many respects. The use of topological methods and an emphasis on the relationship between the continuous and the discrete, via an interplay between geometry and graphical constructions are the highlights of this work. It contains a lot of ideas that to my knowledge, have not been explored anywhere else for studying networks and multi-agent systems. It is my hope, that this work will be a good launching pad for exploring ideas further in this direction, both for myself and for our community of research.

Personally, this thesis has given me an opportunity (as well as as excuse) to learn about several other fascinating subjects, ranging from some deep problems in systems theory and control to excursions into some of the most fundamental topics in mathematics. It has opened for me a window into the abstract but beautiful world of topology, geometry and mathematical analysis. It is my hope that the mathematical techniques learned during the course of this thesis will help me to study many more ideas in applied mathematics and engineering, even outside the scope of multi-agent coordination.

Organization of the Thesis

The outline of this thesis can be understood from the following block diagram. The arrows indicate the logical flow in the theory developed in various chapters.



Statement of Contributions

A brief overview of the main contributions of this work is listed below.

- Introduction of connectivity graphs as models of local interactions.
- A systematic study of the space of connectivity graphs, from a geometric and topological point of view.
- Some results on their realizations in their respective configuration spaces.
- A geometric encoding of the connectivity graphs, suitable for producing triangulations.
- Complexity analysis of networks from the point of view of intrinsic structural complexity.

- Recognition of the various topological spaces in networks, as induced from their connectivity graphs.
- A study of various trajectories on the space of connectivity graphs by the movement of agents in the configuration space (the so-called connectivity graph processes).
- The use of semi-definite programming methods for computing feasible transitions in connectivity graph processes.
- The use of cylindrical algebraic decomposition algorithm for the planning of connectivity graph processes.
- A coordinate-free method of solving coverage problems in sensor networks.
- Use of computational topology software in network simulation.
- The application of connectivity graph processes for generating low-complexity formations as well as for collaborative beamforming.
- Identification of several open problems in this area of research.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Magnus Egerstedt, for granting me great freedom and flexibility in my research. Without his support and encouragement, the ideas explored in this thesis would never have materialized. His appreciation for the utility of abstract mathematical ideas in applied sciences has fuelled my own interest in the abstract and helped me to think more clearly on engineering problems. For this I owe him eternal gratitude. I also thank him for carefully going through my publication drafts, correcting my presentations, and helping me learn the various skills of academic research. I thank him for supporting all the travelling and collaborations, which made my student days really enjoyable; for introducing me to various wonderful research themes in systems and control; for introducing me to his outstanding and smart group of friends and colleagues; and for being extremely helpful in all matters academic or otherwise.

Of the various people that helped me with my academic preparation and research, two persons have been extremely kind to me: Margaret Symington at the mathematics department in Georgia Tech, and Robert Ghrist at the University of Illinois at Urbana-Champaign. I wish to thank Dr. Symington for being an excellent class teacher in three semesters of algebraic topology, for the weekly discussions on geometry and topology, for being my advisor for my masters in mathematics, and for being extremely patient with my poor background in this area of mathematics. I also thank her for introducing me to Robert Ghrist. A very special thanks goes to Dr. Ghrist for his hospitality during my two visits to UIUC and for introducing me to the various concepts and techniques in algebraic topology, particularly in relation to networked multi-agent systems. The visit to UIUC in the summer of 2004 was perhaps the most valuable experience in my graduate studies.

I also wish to thank all those faculty members in the electrical engineering, physics, mechanical engineering, and mathematics departments at Georgia Tech, who taught me some wonderful theorems and great engineering. My special thanks to Dr. Erik Verriest

whose class on optimal control was an absolute joy, and who I have turned to many times for guidance and discussion. I also thank him for agreeing to be part of my dissertation committee, reading this manuscript very carefully and for his invaluable suggestions for improvement. I also thank other members of my dissertation committee: Dr. Allen Tannenbaum and Dr. Ayanna Howard. I wish to thank Dr. George Vachtsevanos for giving me graduate research assistantship during my first semester at Georgia Tech. I specially thank the mathematics faculty at Georgia Tech, in particular Stavros Garoufalidis, Igor Belegradek, Joseph Landsberg and Xu-Yan Chen, for being patient with me, and for not letting me feel like an outsider.

Many thanks to Vin de Silva, for hosting my visit at Stanford, and for discussing his ideas on computational topology. Many of those discussions are still fresh in my memory and have been invaluable for generating new ideas. I also thank him and Dr. Gunnar Carlsson for letting me speak at the topology seminar at Stanford.

Many thanks to my current and former lab-mates : Henrik Axsselson, Meng Ji, Florent Delmotte, Tejas Mehta, Mohamed Babaali, Shun-ichi Azuma, Dave Wooden, Anders Gustafsson, Mauro Boccadoro, Christopher Alvino, Jeremy Jacksonn, and Ganesh Sundaramoorthi. I have enjoyed their company during lunches, discussions, group meetings, and strolls through the campus. Special thanks to my former office-mate and friend, Irtaza Barlas, for helping me settle down during my first days at Georgia Tech.

I wish to thank Mehran Mesbahi at the University of Washington, for discussing dynamic graphs and graph controllability; Ali Jadbabaie at the University of Pennsylvania, for valuable discussions on Euclidean distance matrices among other things ; and Ulf Jönsson at the Royal Institute of Technology, Sweden, for discussing the S-procedure. I also wish to thank Islam Hussain, at University of Illinois at Urban-Champaign, for hosting me during a visit to the Coordinated Science Laboratory (CSL) in UIUC, and for his interesting discussions on cooperative control. Many thanks to Muhammad Sabieh Anwar (currently with the University of California at Berkeley), for giving an intellectually stimulating company during my visit to Caltech.

I would also like to acknowledge the help and concern of the administrative staff at the

electrical engineering department. I am especially thankful to Ms. Jalisa Norton, Ms. Pam Halverson and my academic advisor Ms. Marilou Mycko.

Also, many thanks to my former roommates and fellow Georgia Tech students : Shabbir Husain, Hasan Abbasi, Faisal Shah, Yasser Bhatti, Attir Khalid, Ahmed Usman, Amil Haque, Adnan Ahmed, Mustaeen Naeem and Nafees Tahir. I am extremely thankful to the Pakistani community at Georgia Tech, specially the graduate students and their families for all the support and warmth. My special thanks (in no particular order) to Faisal Khan, Adeel Khalid, Murtaza Askari, Rehan Saeed, Haris Rasheed, Taimoor Khwaja, Messam Naqvi, Salman Aslam, Ali Hashmi, Nadeem Shafi, Jahanzeb Burki, Tauseef-ur-Rehman, Irtaza Shah, Manzar Abbas and their families. All of them made me feel at home, away from home. Thankyou!

I also thank my old friend and former colleague Ahtasham Ashraf, whose advice of not joining a PhD program was completely ignored by me, and who ignored my advice to go for a PhD, by pursuing a successful career in industry and entrepreneurship. I thank him for keeping regular contact on the phone, paying many visits, and for talking incessantly on anything from research problems in sensor networks and signal processing to business and politics.

I would also like to mention the support of my friends and former colleagues at And-Or Logic and the Advanced Engineering Research Organization (AERO), both in Islamabad, Pakistan. My work there preceded my graduate studies, and helped me to get an appreciation for the complexity of real-life engineering problems. My special thanks to Ali Irfan, Muhammad Ali, Fayyaz Gilani, Imran Raouf and Muhammad Amer, for still keeping in touch and for providing updates on the state of the industry back home.

I would like to acknowledge the concern and support of all the friends and colleagues during the weeks that I was stuck in Pakistan for my US-visa renewal, and during the emotional upheavals experienced in my student days in the US that witnessed the 9/11 attack, two senseless wars, a nuclear standoff between India and Pakistan, the death of my loving grandfather, and the recent earthquake in Pakistan.

My last thoughts go to my parents in Pakistan. I thank them for inculcating in me the

pursuit of learning and scholarship. Most of all, I am eternally grateful to them for their loving affection and prayers. The integrity of my father and the meticulous hardworking nature of my mother have been a guiding light for me throughout my life. I am thankful to God for blessing me with such loving parents. My love and thanks also goes to my brother, who has taken it on himself to love and care for every sick and elderly in the family. With him around, my long absence from home should not have been felt.

Last and not the least, my most affectionate thoughts go to my wife, Maria. We got married in the middle of my PhD program at Georgia Tech and she gracefully accepted all the hardship and turmoil of the student life. It was her who bore the burden of my workload, busy weekends, never-ending publication deadlines, long assignments and an empty bank account. I thank her for appreciating the value of my work, and for believing in my judgement to pursue this degree despite all the hardship. Words cannot express my appreciation for providing the comfort of a good home, wonderful food and great companionship. Our thanks to her parents and other members of our families for their continual support and affection.

Abubakr Muhammad,

Atlanta,

Ramadhan, 1426 AH.

TABLE OF CONTENTS

CHAPTER I

BACKGROUND AND INTRODUCTION

The problem of coordinating multiple autonomous agents has attracted significant attention in recent years [8, 12, 38, 53, 57, 67, 71, 91, 96]. Due to advances in many enabling technologies such as advanced communication systems, novel sensing platforms, and cheap computation devices, the realization of large scale networks of cooperating mobile agents has become possible. The research in this area is driven by several commercial and military applications of mobile sensor networks and distributed robotics as well as inspiring links to research problems in ecology, social behavior, statistical physics, bio-informatics, and computer graphics [12, 39, 67, 58, 90, 85]. The surge of interest in this area can be appreciated by the increase in the number of research articles, workshops, conferences, seminars, and funding opportunities dedicated to this subject, and by the enthusiastic involvement of researchers in computer science, engineering, mathematics, biology, and other disciplines. This research is being done under the banners of networked control systems, cooperative control, swarm systems, multi-agent control, distributed robotics, etc.

All such research efforts are focused on understanding, modelling, and employing four basic ingredients of a networked control system : *sensing, communication, computation,* and *control*. It is important to point out that this research area is distinguished from some of its sister subjects, where one or more of the four ingredients are of little or no relevance. These areas include sensor networks (sensing, communication, and computation), parallel and distributed processing (communication and computation), multi-body dynamics (sensing and control), distributed control (sensing, computation and control) etc. Therefore the design and study of multi-agent robotic systems demands the attention of researchers that are cognizant of all four ingredients. It is due to this unique confluence of different areas that several researchers, especially in the systems and control community, have taken the opportunity to enrich the understanding of their own subjects while making contributions

to an area that is rapidly becoming a subject in itself.

The systems and control community in particular, has been attracted to this area. There are many interesting theoretical problems that can be readily modelled and solved using conventional techniques in systems theory. On the other hand, several open problems in networked control systems are fuelling research into the development of techniques that involve a modification of the classical techniques through interactions with other disciplines in mathematics and applied sciences. This has resulted in a tremendous enrichment of control theory. The interdisciplinary nature of this research has helped define several areas inside control theory such as cooperative control, hybrid systems, and networked control; and spawned a number of applications. A glimpse of this synergy can be seen in this work, where ideas from graph theory and, perhaps for the first time, from algebraic topology have been used to study network structures.

In the study of networked control systems, sensory and communication constraints limit the effectiveness of the local interactions, whereas the absence of hierarchial structures combined with scalability issues make global coordination a difficult and complex problem. One of the main challenges in designing such systems is to provide a rigorously proven framework of decentralized coordination that overcomes these limitations. In this chapter, we summarize the state of the art in the design of multi-agent control systems with limited local interactions. We argue that the study of geometric and topological structure of networks is essential for the effective study and design of such systems. We also emphasize that the results of this research is likely to benefit several areas outside robotics such as social and biological multi-agent systems, where similar issues in decentralized coordination are poorly understood.

1.1 From Single- to Multi-agent Robotics

The control and design of robotic systems is a well-studied subject. Due to its numerous applications, most problems involving a single robotic system have been analyzed in great detail by various researchers. Lately, a lot of focus has been placed on designing autonomous systems that require minimum human intervention and can execute their tasks in uncertain

environments. The reliability and performance of single autonomous robotic systems have increased considerably due to advances in AI, mechatronics and control theory. On the other hand, there has been a substantial reduction in their cost and size due to developments in various enabling technologies. During the last decade, many researchers have realized that it is feasible to deploy a large team of collaborating robots to execute tasks that are impossible or difficult for a single system. The underlying driver of these research efforts is the implicit assumption that there is strength in numbers, which can be exploited when exploring and negotiating unknown or hostile environments.

Some of the techniques developed for single agents, interacting in both structured and unstructured environments, such as trajectory tracking, nonlinear control, mapping and localization, are readily applicable in the multi-agent case as well. Promising applications such as space exploration, cooperative control of unmanned aerial vehicles (UAV's), advanced manufacturing, mobile sensor networks, coordinated building, etc. [12, 39, 58, 67, 85, 90], have driven researchers to develop a framework for multi-agent systems from a system theoretic point of view. However, a number of challenges stemming from the distributed and hence local nature of the information available to the individual agents in the formation, have presented themselves.

1.2 Inspiration from Nature

In order to look for inspiration when trying to model such systems, roboticists have increasingly begun to look to naturally occurring systems, where distributed multi-agent systems are abundant [28, 77]. These systems range from human societies, where each agent is an extremely complex system in itself and the social behavior transcends beyond simple mechanical tasks, to lifeless physical systems made of particles, atoms or molecules [9]. The latter carry no intelligence themselves, but interact using simple physical laws, and give rise to complex adaptive systems as a group . Robotics finds its place somewhere in between these two extremes. Needless to say, the comprehension of the complicated behavior of human societies may be the ultimate goal for a multi-agent system designer, but it is far too difficult and exists only in science fiction at this point. The state of the art in multi-agent

robotics is instead challenged by more humble objectives like terrain exploration, coordinated building and manipulation, planning of team formations, etc. On the other hand, inspiration from lifeless physical systems is inadequate, as most types of interactions (e.g. potential fields) in physical systems can be simulated in hardware without the use of large computing power, sensing and communication capabilities carried by modern robotic platforms. The best "role-model" for multi-agent robotics can be found in biological systems where group behavior is manifested in schools of fish, insect swarms, animal herds, bacterial colonies, bird formations, etc . There is a remarkable similarity between the group behaviors found in biological systems and the ones that roboticists want their artificial systems to exhibit. Similarly, a lot can be learnt from artificial robotic systems when modelling behaviors for animal groups.

1.3 Centralized Vs Decentralized Multi-agent Systems

Most problems in multi-agent robotics revolve around defining formations and developing control laws that guarantee formation stability. (See for example [12, 34, 39, 40, 62, 85, 95].) In most cases, the underlying assumption is that each individual robot has complete knowledge of the whereabouts of all other robots. This is a *centralized* way of approaching coordination problems. However, several practical issues make this assumption false. In particular, if the number of robots is large, bandwidth limitations as well as range constraints on the available sensing capabilities, imply that the global knowledge assumption has to be abandoned. Hence, recent work in multi-agent robotics has emphasized *decentralized* coordination to maintain formations [12, 34, 35, 38, 91, 97].

As the shift is made from global to local interactions, there are a number of issues that need to be resolved, stemming from the inherent global nature of a formation. For instance, if a number of robots have to decide which individual roles to take on in the formation *decentralized* decision making mechanisms have to be employed. Similarly, different formations are potentially beneficial in different situations. For example, when exploring unknown terrains, maximally spread formations may be preferable, but as obstacles are encountered new formations must be used, e.g. for negotiating narrow passages [39]. As of yet, little

work has been done on how to choose formations in a decentralized and autonomous fashion as a reaction to changes in the environment.

1.4 Swarming, Flocking and Formation Control

The problem of decentralized control has been most successfully addressed when investigating swarm behaviors, where the individual robots are moving according to limited range potential fields (e.g. [40, 90]), or according to some averaging orientation rules [53, 98]. However, these results are not *constructive* in the sense that one can not specify or change desired formations in any direct manner. Therefore the subject of *formation control* should be distinguished from studies on swarming and flocking behaviors. Although formation control has been a favorite subject among multi-agent roboticists for many years, it has not progressed beyond simple formations for a small number of agents. The most obvious reason for this is the failure to recognize the importance of communication among agents. This may have happened because until recently, researchers did not have the luxury of equipping a large number of robots with advanced communication equipment. With the advent of inexpensive high-speed wireless networks, today's networked control systems are complex communication networks that are faced with all issues of managing network traffic. Still, little has been done so far to understand the role of information exchange in multi-agent robotics.

1.5 Limited Communication and Sensory Capabilities

Before one can properly define what is meant by formations or distributed control, a number of *modelling issues* need to be resolved. Among them, a basic problem is how to capture the local nature of the interactions due to inter-robot communication and sensory perception, when characterizing multi-agent formations. In most applications of networked control systems, the individual robots can collect information about their environment and neighboring robots in the formation by either peer to peer communication or by relying on sensory information. Since any physical sensor is limited by its range, resolution, and calibration errors, the information available to each agent by either direct observation or state

estimation is always limited and uncertain. In non-omnidirectional sensors, the limitations may also arise due to the directivity patterns of sensors. Examples include the conic field of view of a camera and the radiation patterns of antennas and sonars [63].

Instead, if the robots share information using peer to peer communication strategies, this problem can be overcome as long as the number of robots in the formation is relatively small. However, as the formation size increases, both in cardinality and spatial dimension, several factors such as bandwidth limitations, large distances, and corruption in communication channels due to noise and interference severely limit the possibility to convey and use global information. A good algorithm for multi-agent coordination should be scalable in terms of the size of the network. Therefore, no individual robot can be assumed to have complete knowledge about the states of every other robot of the formation. This restriction leads to the question of how local interactions should be represented.

1.6 Spatial Relationships in Formations: Graph Theoretic Models

One natural way in which local interactions can be expressed is to let this aspect of the formation be represented as a graph, in which nodes correspond to individual agents and the presence of an edge between two nodes (robots) signifies that an interaction exists between them. In other words, an edge between two nodes implies that either the corresponding robots are within sensory range of each other, or that a communication channel is established between them. These types of graphs have been proposed in the literature to represent spatial or geometric relationships between agents. Examples include [38, 53, 73, 92, 98, 99, 97].

Consider, for example [92], in which Saber et. al. define a *spatial adjacency matrix* for formations of agents, equipped with sensors of limited range as follows: Let $x = (x_1, x_2, \ldots, x_N)$ denote the position vector of N robots in an ambient space $\mathbb{R}^k$, where

$k = 1, 2$ or 3. If agent i has an omnidirectional sensor of range δ_i then, the spatial adjacency matrix $A(x) = [a_{ij}]$ is defined as

$$a_{ij}(x) = \begin{cases} 1 & \text{if } x_j \in \mathcal{B}_{\delta_i}(x_i),\ j \neq i, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathcal{B}_{\delta_i}(x_i)$ is the closed ball in $\mathbb{R}^k$ with radius δ_i, centered at x_i.

If the sensors are directional, or if the agents themselves have an orientation, conic neighborhoods can be used to define the spatial adjacency matrix as

$$a_{ij}(x) = \begin{cases} 1 & \text{if } \|x_j - x_i\| \leq \delta_i,\ |\theta_j - \theta_i| \leq \phi_i,\ j \neq i, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where θ_i denotes either the orientation of robot i, or the directionality of its sensor. Moreover, ϕ_i defines the conic neighborhood in which robot i can effectively acquire information from neighboring robots. It should be noted that these models (especially in the conic case) imply directed rather than undirected graphs. In other words, the relationships that model the graphs need not be commutative.

These types of constructions capture the local interactions in a straight forward manner. Recently, many researchers have explored and characterized the graph theoretic properties associated with these graphs. Some striking results have been obtained using *algebraic graph theory* [45]. In [38], the stability of a formation has been shown to be closely related to the *Laplacian* L of the underlying graph. Let the number of edges incident on a node v be denoted by $deg(v)$. If we define the *degree matrix* Δ of a graph as

$$\Delta_{ij} = \begin{cases} deg(v_i), & i = j, \\ 0, & i \neq j, \end{cases} \tag{3}$$

then the *Laplacian* L is defined by:

$$L = \Delta - A, \tag{4}$$

where A is the adjacency matrix. The rank of L is related to the connectivity of the graph [45]. In [92], similar results have been obtained for the construction of agreement protocols between the robots. In [53, 98, 99], the Laplacian has been used for studying the alignment

of individual robots based on nearest neighbor averaging rules. It should be emphasized again, that none of these methods are constructive and mostly concern qualitative notions like swarming and flocking.

1.7 Configuration Spaces: Lessons from Robotic Manipulators

In the previous paragraphs, it has been shown how graph theoretic models lend themselves well to capturing certain aspects of spatially induced relations between robots in a multi-agent system. Another aspect of studying multi-agent formations that obey spatial relationships, is to obtain the configuration space of the formation. Even if the agents are fully actuated and living in Euclidean spaces, the configuration space of formations is non-trivial. In the multi-agent robotics literature, it is quite standard to assume that the robots are evolving on the simplified Euclidean configuration space, namely the product space $\mathbb{R}^k \times \mathbb{R}^k \times \ldots \times \mathbb{R}^k = (\mathbb{R}^k)^N$. However, as pointed out by Ghrist in [41], the configuration space of N robots, even without any inter-agent constraints is

$$C^N(\mathbb{R}^k) = (\mathbb{R}^k \times \mathbb{R}^k \times \ldots \mathbb{R}^k) - \Delta, \tag{5}$$

where $\Delta = \{(x_1, x_2, \ldots, x_N) : x_i = x_j \text{ for some } i \neq j\}$. For many applications it is also required that the configuration be independent of the relative labelling of the different agents in a formation. In such cases one should consider the space obtained by the action of the permutation group $\mathcal{S}_N$ on the space $C^N(\mathbb{R}^k)$, namely the space $C^N(\mathbb{R}^k)/\mathcal{S}_N$. When inter-agent constrains are present and are represented as a graph, one could ask how the configuration evolves while the graph is preserved. Since the movements of the individual agents make the graph a dynamic structure as well, the characterization of a fixed spatial relationship between the robots in the form of a configuration space is useful.

The concept of configuration spaces is well-understood for robotic manipulators and multi-body systems, where the kinematics are studied for flexible and rigid bodies obeying certain physical laws. The configuration spaces arising in such systems are usually smooth manifolds such as Lie groups that have nice properties [83]. Configuration spaces have also been studied by mathematicians as *algebraic varieties* [48] corresponding to mechanical

linkages. For example, in [70] and [54], the configuration space of a weighted graph corresponding to the mechanical linkages between joints in the manipulator has been described as the set of all possible realizations of the graph. Recently, some *universality theorems* have also appeared in the literature [56] that answer the converse problem of whether a mechanical linkage exists for a given algebraic variety. For multi-agent formations defined by spatial and geometrical relationships, this is a very narrow point of view, although it is fair to say that many researchers have produced some good studies by following that approach. In some works, such as [35, 85, 91], researchers have looked at formations as rigid bodies and modelled their behavior as N-particle systems driven by artificial potential functions. Such methods are however restrictive and do not take full advantage of the extra flexibility in the coupling among the robots. Therefore, it is clear that this issue cannot be addressed using graph theory or rigid-body mechanics alone.

Since many geometric constraints are polynomial, the resulting configurations spaces can in many cases be described as *semi-algebraic sets*. However, the problem of determining the configuration space of a graph defined by inequality constraints (instead of equality constraints in rigid mechanical linkages) is considerably difficult. If such a characterization is properly understood, this may lead to utilizing the well-established methods of designing vector fields on configuration spaces and deriving control laws.

The conclusion to be drawn from the research efforts mentioned above is that there is a need to develop a rigorous framework for totally decentralized coordination algorithms with incomplete global information. Such coordination algorithm should be constructive and not limited to studying emergent behaviors like swarming and flocking. A number of modelling challenges have to be overcome for perception and communication limitations. The role of information exchange is still very poorly understood. A number of researchers have employed graph theoretic models, but there is a lack of understanding about the type of graphs that arise in cooperative control problems. Finally, the role of configuration spaces is little understood in this framework.

CHAPTER II

FROM GRAPH THEORY TO ALGEBRAIC TOPOLOGY

The field of algebraic topology has a fascinating history of success in applied sciences. At the end of the 19th century, when Henri Poincaré pioneered many of the computational tools used in topology today, he immediately put them to use for studying dynamical systems. Physicists were quick to utilize these tools in quantum mechanics, gravitation, string theory, and for studying crystalline structures. Computer scientists have recently begun using topology for analyzing distributed algorithms. In this thesis, a significant departure has been made from the different approaches that are currently in use for the analysis and design of networked control systems. The main highlight of this work is the use of algebraic topology, in addition to the usual graph theoretic methods for analyzing networks.

Although algebraic topology is a relatively lesser known branch of mathematics in the control systems community, the application of topological tools in systems theory is not new. Its sister subjects of differential and algebraic geometry have found wide application in control, robotics and optimization. The theory of geometric control [83] relies heavily on the topological properties of the underlying manifolds that form the configuration spaces of the systems under study, although this relationship has often been overlooked or taken for granted. Similarly the qualitative theory of dynamical systems, pioneered by Poincaré, has many topological overtones and is widely used in the systems and control community without an explicit reference to the underlying concepts in topology. A small community of researchers [1, 3, 22, 36, 37, 41, 44] has emphasized this connection over the last few years. However, the abstract nature of this subject has prevented its propagation in the controls systems community at-large.

Figure 1: To a topologist, both the coffee mug and the doughnut are the same.

2.1 What is Algebraic Topology?

Informally, algebraic topology is the study of spaces and maps between them using algebraic methods [5, 49]. The shapes and spaces may represent circles, spheres, doughnut-shaped figures, or something more exotic like Klein bottles or Mobius bands. These are all examples of topological spaces. One of the basic objectives of algebraic topology is to find invariants of these topological spaces. These invariants should be able to distinguish between different shapes. The invariants are algebraic objects that can be computed from a representation of the topological space. The most important topological invariants are the so-called homotopy and homology groups. It turns out that these invariants distinguish between topological spaces by essentially measuring the number of holes in a topological space.

There is an interesting saying that a topologist is a person who cannot tell a coffee mug from a doughnut. A doughnut is a surface with one hole in it. So is a coffee mug, where the handle is the hole. Imagine that the doughnut is made of stretchable rubber or clay. Then one can stretch and deform the doughnut to get a coffee mug without creating or closing any holes. (See Figure 1). This theme of distinguishing topological spaces by 'counting holes' will appear in several applications, later in this thesis.

2.2 Emergence of Computational Algebraic Topology

During the last few years, methods of algebraic topology have proliferated into many areas of applied sciences. One of the main reasons for this surge of interest in topology is the availability of computational packages that allow fast computations on large sets of data and require minimal knowledge of the underlying abstract concepts in topology. The area of computational topology, itself is an emerging field of applied mathematics with several

opportunities of research. A good survey of this research can be found in [13, 55].

Some of the most difficult and least understood issues in geometric computing involve topology. The aim of computational topology is to go beyond the state of the art in computational geometry with an emphasis on provable correctness, efficiency, and robustness to continuous domains, curved surfaces, and higher dimensions. Such an extension brings computational geometry into contact with classical topology for shape representation, manipulation, and analysis.

Topology separates global shape properties from local geometric attributes, and provides a precise language for discussing these properties. Such a language is essential for composing software programs, such as connecting a mesh generator to a computational fluid dynamics simulation. Many software packages have emerged that specifically deal with such issues. Out of these, two packages `CHomP` and `Plex` [23, 89] have been used in this thesis.

2.3 Topology for Networked Control Systems

As emphasized in the previous chapter, an important theme in the study of such systems is the use of graph-theoretic models for describing the local interactions in the network. These graph-based models can serve as a bridge between the continuous and the discrete when trying to manage the design complexity associated with formation control problems. A recurring theme in many of these works has been the use of algebraic graph theory for incorporating network structure into the equations of the coupled dynamical system.

In this work, it has been realized that the discrete nature of this graphical representation can be further generalized to combinatorial objects, known as simplicial complexes, that are extremely familiar in the field of algebraic topology. This empowers us to work at an abstraction level which is higher than algebraic graph theory. For example, the graph Laplacian mentioned above arises naturally from the matrix representation of the so-called boundary operators on simplicial complexes. Based on this realization, we have at our disposal a whole arsenal of results that have been developed in the last century for studying these abstract objects.

The motivation for using these topological tools also stems from the need to design

large scale multi-agent systems which achieve global objectives using local interactions in a completely decentralized manner. Moreover, the topological abstraction of networks simplifies the solution to a number of problems that are normally tackled with a geometrical insight only. Finally, the availability of computational topology software like `Plex` and `CHomP` [23, 89] makes it possible to explore these tools for large systems with a minimal knowledge of abstract concepts in topology. It is the author's belief that the emerging area of networked control systems, along with recent advances in computational algebraic topology have set the stage for this subject to appeal to wider audiences in systems and control.

2.4 Why and When to Use Algebraic Topology?

A valid question that can be asked about the use of topological methods for networked control systems is the following: What is the advantage of using the 'language' of algebraic topology? Can we present the same results using the usual graph theory and some statements on geometry? Surely, the language of algebraic topology is very elegant, and once this language is learnt, the entire sophisticated machinery of topology becomes accessible for tackling a whole range of other problems. But have we produced some results that *can not* be obtained using graph theory and geometry alone? If one carefully studies the results in Chapters 6 and 8, they are indeed statements of graph theory, combined with some observations on geometry. But, the work presented in Chapter 11, in particular the results based on higher homology groups and the deep statements of Čech cohomology theory are too complicated to be expressed in any other way. Therefore, algebraic topology is not merely a mechanical tool for doing proofs, rather it empowers us with some very deep concepts. Concepts such as homology, homotopy and cohomology are very general ideas and principles [43].

A final word of caution: It is not the intention of the author to present computational algebraic topology as a panacea for every problem in networked control systems. Rather, it is suggested that there is a broad class of problems in multi-agent coordination, which can be handled in a natural way by dealing with various topological spaces associated with

multi-agent networks. As evident from a large part of the work presented in this thesis, there are quite a number of problems in multi-agent coordination, that can be best handled using more familiar tools of analysis and design, such as graph theory and computational geometry. The topological methods are only applied when necessary, and where they give a significant advantage over traditional methods.

PART I

Static Aspects: The Structure of Connectivity Graphs

CHAPTER III

CONNECTIVITY GRAPHS AND THEIR REALIZATIONS

In many multi-agent coordination problems, a finite representation of the configuration space appears naturally, namely by using graph-theoretic models for describing the local interactions in the formation. In other words, graph-based models can serve as a bridge between the continuous and the discrete when trying to manage the design-complexity associated with formation control problems. As explained in the previous chapter, different types of graphs are used for modelling which neighboring robots a given robot can communicate with. In this chapter, we show that the graphs that can represent formations do in fact correspond to a proper subset of all graphs, denoted by the set of connectivity graphs. This motivates a systematic study of the relation between graphs and their configuration spaces, and in particular, the problem of realizability of a graph in the corresponding configuration space.

3.1 Formations and Connectivity Graphs

Graphs can model local interactions between agents, when individual agents are constrained by limited knowledge of other agents. In this chapter we present a graph theoretic formalism for describing formations in which the primary limitation of perception for each agent is the range of its sensor. Suppose we have N such agents with identical dynamics evolving on $\mathbb{R}^2$. Each of the agents carries a pre-assigned unique identification tag $n \in \{1, 2, \ldots, N\}$. Each agent is also equipped with a range limited sensor by which it can sense the position of other agents. All agents have identical sensor ranges δ. Let the position of each agent be $x_n \in \mathbb{R}^2$, and its dynamics be given by

$$\dot{x}_n = f(x_n, u_n),$$

where $u_n \in \mathbb{R}^m$ is the control for agent n and $f : \mathbb{R}^2 \times \mathbb{R}^m \to \mathbb{R}^2$ is a smooth vector field. The configuration space $\mathcal{C}^N(\mathbb{R}^2)$ of the agent formation is made up of all ordered N-tuples in $\mathbb{R}^2$, with the property that no two points coincide, i.e.

$$\mathcal{C}^N(\mathbb{R}^2) = (\mathbb{R}^2 \times \mathbb{R}^2 \times \ldots \mathbb{R}^2) - \Delta,$$

where $\Delta = \{(x_1, x_2, \ldots, x_N) : x_i = x_j \text{ for some } i \neq j\}$. The evolution of the formation can be represented as a trajectory $\mathcal{F} : \mathbb{R}_+ \to \mathcal{C}^N(\mathbb{R}^2)$, usually written as $\mathcal{F}(t) = (x_1(t), x_2(t), \ldots x_N(t))$ to signify time evolution. The spatial relationship between agents can be represented as a graph in which the vertices of the graph represent the agents, and the pair of vertices on each edge tells us that the corresponding agents are within sensor range δ of each other (See Figure 2). However several formations may give the same graph. We make these ideas precise as follows.

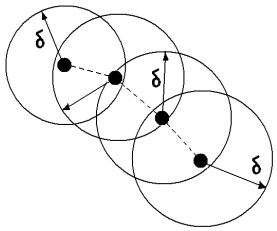


Figure 2: Agents and their connectivity graph.

Definition 3.1.1 (Connectivity Graph of a Formation) *Let $\mathcal{G}_N$ denote the space of all possible graphs that can be formed on N vertices $V = \{v_1, v_2, \ldots, v_N\}$. Then we can define a function $\Phi_N : C^N(\mathbb{R}^2) \to \mathcal{G}_N$, with $\Phi_N(\mathcal{F}(t)) = \mathcal{G}(t)$, where $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t)) \in \mathcal{G}_N$ is the* connectivity graph *of the formation $\mathcal{F}(t)$. $v_i \in \mathcal{V}$ represents agent i at position x_i, and $\mathcal{E}(t)$ denotes the edges of the graph. $e_{ij}(t) = e_{ji}(t) \in \mathcal{E}(t)$ if and only if $\|x_i(t) - x_j(t)\| \leq \delta$ $i \neq j$. In other words,*

$$\begin{aligned} \Phi_N(\mathcal{F}(t)) = &(\{v_1, \ldots v_N\}, \{(v_i, v_j) \mid i \neq j \text{ and} \\ &\|x_i(t) - x_j(t)\| \leq \delta\}) \end{aligned} \tag{6}$$

Some observations about these connectivity graphs can be made already at this point.

- The graphs are *simple* by construction i.e. there are no self-loops or parallel edges between the same two nodes.
- The graphs are always undirected because the sensor ranges are identical.
- The motion of agents in a formation may result in the removal or addition of edges in the graph. Therefore $\mathcal{G}(t)$ is a dynamic structure.
- Every graph in $\mathcal{G}_N$ is not a connectivity graph.

The last observation is not as obvious as the other. While many researchers have referred to graphs of formations as their models; [92, 53]; the issue of whether an arbitrary graph corresponds to a proper agent formation has been mostly overlooked. We discuss this point in detail.

3.2 Every Graph is Not a Connectivity Graph

We start with the following definition.

Definition 3.2.1 (Realization of a graph in $\mathcal{C}^N(\mathbb{R}^2)$) *A* realization *of a graph $\mathcal{G} \in \mathcal{G}_N$ is a formation $\mathcal{F} \in \mathcal{C}^N(\mathbb{R}^2)$, such that $\Phi_N(\mathcal{F}) = \mathcal{G}$.*

An arbitrary graph $\mathcal{G} \in \mathcal{G}_N$ can therefore be *realized* as a connectivity graph in $\mathcal{C}^N(\mathbb{R}^2)$ if $\Phi_N^{-1}(\mathcal{G})$ is nonempty. We denote by $\mathcal{G}_{N,\delta} \subseteq \mathcal{G}_N$, the space of all possible graphs on N agents with sensor range δ, that can be realized in $\mathcal{C}^N(\mathbb{R}^2)$. Let us start by analyzing this space for small values of N. For $N = 1$, the configuration space is $\mathcal{C}^1(\mathbb{R}^2) \simeq \mathbb{R}^2$ and the only possible graph on one agent is always realizable. For $N = 2$, the situation corresponds to whether the two agents are within δ distance of each other or not. Therefore all formations in the subset $\{(x_1, x_2) : \|x_1 - x_2\| \leq \delta, x_1 \neq x_2\} \subset \mathcal{C}^2(\mathbb{R}^2)$ correspond to the connected graph of 2 vertices, while the remaining configuration space corresponds to the situation when the graph is disconnected.

If we now move on to the case for $N = 3$ similar constructions can be obtained for various connected and disconnected graphs on 3 vertices. Consider for example the situation in Figure 3, where the 3 agents are positioned at the points marked by circles. Let each

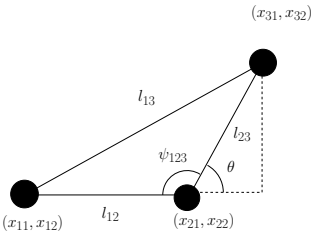


Figure 3: Depicted are three robots and their inter-robot distances.

position x_i be given by its Cartesian coordinate pair $(x_{i1}, x_{i2})^T$. For notational convenience let $\|x_1 - x_2\| = l_{12}$, $\|x_2 - x_3\| = l_{23}$ and $\|x_1 - x_3\| = l_{13}$. Also let θ and ψ_{123} be the angles shown in the figure. In general, any connectivity graph on N vertices imposes various constraints on the relative positions of individual agents in the configuration space $\mathcal{C}^N(\mathbb{R}^2)$. In the case of a connected graph on 3 vertices, the constraints on positions x_1, x_2 and x_3 correspond to a single constraint on the angle ψ_{123}, when the the agents are positioned as shown in Figure 3. This simple observation will subsequently lead to some interesting properties of the connectivity graphs and their realizations. Suppose we are considering the line graph on 3 vertices in Figure 5, then the given geometrical configuration corresponds to this graph if $l_{12} \leq \delta, l_{23} \leq \delta$, and $l_{13} > \delta$. Moreover we can write

$$\begin{aligned} l_{13}^2 &= (l_{12} + l_{23}\cos\theta)^2 + (l_{23}\sin\theta)^2, \\ &= l_{12}^2 + l_{23}^2 + 2l_{23}l_{12}\cos\theta. \end{aligned}$$

If $l_{13} > \delta$ then

$$\cos\theta > \frac{\delta^2 - l_{12}^2 - l_{23}^2}{2l_{23}l_{12}}.$$

It is easy to see that the term on the right has a minimum corresponding to the maximum values of $l_{12} = l_{23} = \delta$. Therefore $\cos\theta > -\frac{1}{2}$ which means that $\theta \in [-\frac{2\pi}{3}, \frac{2\pi}{3}]$. Therefore the smaller angle between l_{12} and l_{23} satisfies $\psi_{123} = \pi - \theta > \frac{\pi}{3}$, for all $0 < l_{12}, l_{23} \leq \delta$ and $l_{13} > \delta$. Hence, whenever we have two edges e_{ij} and e_{ik} in a connectivity graph that share a vertex v_i in such a way that there is no edge between vertices v_j and v_k, then

$$\psi_{j,i,k} \triangleq \cos^{-1}\left(\frac{< x_j - x_i, x_i - x_k >}{\|x_j - x_i\|\|x_i - x_k\|}\right) > \frac{\pi}{3} \tag{7}$$

Now, denote by S_N the "star graph" in $\mathcal{G}_N$ i.e. the graph which has $N-1$ vertices $v_2, v_3 \ldots v_N$ of degree 1 and one vertex v_1 with degree $N-1$. An example of such a graph

is shown in Figure 4.a. Also denote by $\Diamond_5$ and $\Diamond_6$, the graphs in $\mathcal{G}_5$ and $\mathcal{G}_6$ respectively, as drawn in Figures 4.a and 4.b.

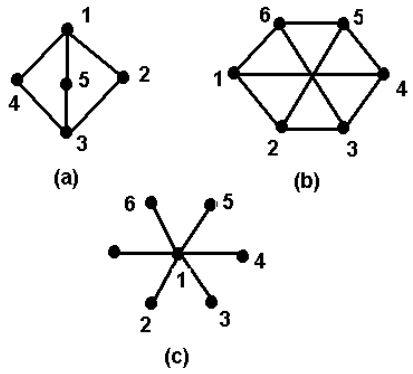


Figure 4: Graphs $\Diamond_5$, $\Diamond_6$ and S_7, that are not connectivity graphs

Proposition 3.2.1 *The graphs $\Diamond_5 \in \mathcal{G}_5$ and $\Diamond_6 \in \mathcal{G}_6$ do not belong to $\mathcal{G}_{5,\delta}$ and $\mathcal{G}_{6,\delta}$ respectively.*

Proof: Suppose that to the contrary $\Diamond_5 \in \mathcal{G}_{5,\delta}$ then there exists some realization $\mathcal{F} = (x_1, x_2, \cdots x_5) \in \mathcal{C}^5(\mathbb{R}^2)$ such that $\Phi_5(\mathcal{F}) = \Diamond_5$. From Equation 7 it follows that the angles $\psi_{415},\ \psi_{512},\ \psi_{123},\ \psi_{235},\ \psi_{534}$ and ψ_{341} are all greater than $\frac{\pi}{3}$. Therefore,

$$\psi_{415} + \psi_{512} + \psi_{123} + \psi_{235} + \psi_{534} + \psi_{341} > 6\left(\frac{\pi}{3}\right) = 2\pi$$

But since $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$ are vertices of a polygon of 4 sides, $\psi_{415} + \psi_{512} + \psi_{123} + \psi_{235} + \psi_{534} + \psi_{341} = 2\pi$ which is a contradiction. Therefore $\Diamond_5 \notin \mathcal{G}_{5,\delta}$.

Similarly if we assume that $\Diamond_6 \in \mathcal{G}_{6,\delta}$ then by a repeat of the above argument we get $\psi_{416}+\psi_{163}+\psi_{365}+\psi_{652}+\psi_{452}+\psi_{145}+\psi_{143}+\psi_{436}+\psi_{632}+\psi_{325}+\psi_{521}+\psi_{214} > 12\left(\frac{\pi}{3}\right) = 4\pi$. However from the condition that $x_1, x_2, \cdots, x_6 \in \mathbb{R}^2$ are vertices of a polygon of 6 sides, this sum should be exactly equal to 4π, which is a contradiction. Therefore $\Diamond_6 \notin \mathcal{G}_{6,\delta}$. ∎

Proposition 3.2.2 *$\mathcal{S}_N \in \mathcal{G}_N$ does not belong to $\mathcal{G}_{N,\delta}$ for $N > 6$.*

Proof: Suppose to the contrary, $\mathcal{S}_N \in \mathcal{G}_{N,\delta}$. If $deg(v_1) = N - 1$, and $(x_1, \cdots x_N) \in \mathcal{C}^N(\mathbb{R}^2)$ is a realization then $\psi_{i,1,j} > \frac{\pi}{3}$ for all $2 \leq i, j \leq N$. We have, $\psi_{2,1,3} + \psi_{3,1,4} + \cdots + \psi_{N-2,1,N-1} + \psi_{N-1,1,N} + \psi_{N,1,2} > (N-1)\pi/3$. If $N > 6$ then this sum is strictly greater

than 2π. However by the given setup, this sum should be exactly equal to 2π. Therefore, by this contradiction $\mathcal{S}_N \notin \mathcal{G}_{N,\delta}$ for $N > 6$. ∎

There are of course many other examples of realizable and non-realizable connectivity graphs. If a graph is completely disconnected, it means that the distance between any two agents in the formation is separated by more than δ. This can easily be achieved by placing the vertices one by one in such a way that x_i does not belong to $\bigcup_{j=1}^{i-1} \mathcal{B}_\delta(x_j)$, where $\mathcal{B}_\delta(x)$ is the closed ball of radius δ centered at x. This observation can be further generalized as follows.

Lemma 3.2.1 *A graph $\mathcal{G} \in \mathcal{G}_{N,\delta}$ if and only if each of its connected component $\mathcal{G}_i \in \mathcal{G}_{M_i}$ is realizable in some $\mathcal{G}_{M_i,\delta}$, $M_i < N$.*

We omit a formal proof here for brevity but the concept is easy to understand. We saw earlier that completely disconnected graphs are trivially realizable by placing the agents further than δ from one another. If $\mathcal{G} \in \mathcal{G}_N$ has many disjoint connected components, say $\{\mathcal{G}_i\}$, we can place each connected component 'far away' from all other components so that none of the agents in one component can sense agents in other connected component. By this construction we have a realization for G if and only if all G_i have realizations in their respective spaces $G_{M_i,\delta}$.

Theorem 3.2.1 *$\mathcal{G}_{N,\delta}$ is a proper subspace of $\mathcal{G}_N$ if and only if $N \geq 5$.*

Proof: In order to prove that $\mathcal{G}_{N,\delta}$ is a proper subspace of $\mathcal{G}_N$ for some N, it is enough to show that $\Phi : C^N(\mathbb{R}^2) \to \mathcal{G}_N$ is not onto. Therefore we need to provide a graph $\mathcal{G} \in \mathcal{G}_N$ such that $\Phi^{-1}(\mathcal{G}) = \emptyset$. From Proposition 3.2.1, we have examples of graphs that are not realizable in $\mathcal{G}_{5,\delta}$ and $\mathcal{G}_{6,\delta}$. For $N \geq 7$ the star graphs $\mathcal{S}_N$ of Proposition 3.2.2 provide the examples of graphs that cannot be realized as connectivity graphs in $\mathcal{G}_{N,\delta}$. This proves that $\mathcal{G}_{N,\delta}$ is a proper subspace of $\mathcal{G}_N$ if $N \geq 5$.

To prove that every graph in $\mathcal{G}_N$, for $N < 5$, is realizable in $\mathcal{G}_{N,\delta}$, we have to enumerate all possible graphs for $N < 5$ and give realizations for each graph. Since we are dealing with a small number $N(< 5)$, the enumeration strategy works well. The number of possible

graphs to check can be further reduced by noting that we need to consider only connected graphs. The justification for this comes from Lemma 3.2.1 given above. In fact, from [19] we know what these graphs are, and they together with their realizations are given in Figures 5 and 6, which completes the proof. ∎

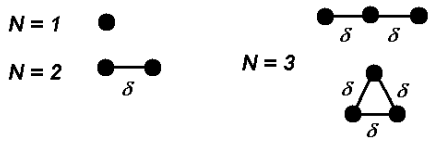


Figure 5: Possible realizations for all $G \in \mathcal{G}_{N,\delta}$ for $N \leq 3$

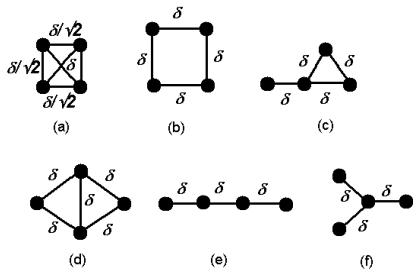


Figure 6: Possible realizations for all connectivity graphs in $\mathcal{G}_{4,\delta}$.

Corollary 3.2.1 *If each connected component $\mathcal{G}_i$ of a graph $\mathcal{G} \in \mathcal{G}_N$ belongs to $\mathcal{G}_{M_i}$, $M_i < 5$ then the graph has a realization in $\mathcal{G}_{N,\delta}$.*

Formations can produce a wide variety of graphs for N vertices. This includes graphs that have disconnected subgraphs or totally disconnected graphs with no edges. However the problem of switching between different formations or of finding interesting structures within a formation of sensor range limited agents can only be tackled if no sub-formation of agents is totally isolated from the rest of the formation [58]. This means that the connectivity graph $\mathcal{G}(t)$ of the formation $\mathcal{F}(t)$ should always remain *connected* (in the sense of connected graphs) for all time t. For notational convenience we use $\mathcal{G}^c_{N,\delta} \subseteq \mathcal{G}_{N,\delta} \subseteq \mathcal{G}_N$ to denote the set of all connected graphs of N vertices that satisfy the connectivity condition of the above definition for sensor range δ.

CHAPTER IV

REALIZATIONS AND SEMI-DEFINITE PROGRAMMING

Some results on the realization of connectivity graphs $\mathcal{G}_{N,\delta} \subseteq \mathcal{G}_N$ have been given in the previous chapter. In the arguments for proving these results, some simple geometrical arguments have been used. Although, these arguments give sufficient conditions for certain graphs, they cannot be easily applied to all graphs. In fact, it will be interesting to know the answer to the following question: Given *any* arbitrary graph $\mathcal{G} \in \mathcal{G}_N$, can it be realized as a connectivity graph in $\mathcal{C}^N(\mathbb{R}^2)$?

4.1 Infeasibility and the Positivstellensatz

Recall that each connectivity graph $(\mathcal{V}, \mathcal{E})$ for the formation $(\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N) \in \mathcal{C}^N(\mathbb{R}^2)$ can be described by $N(N-1)/2$ relations of the following form:

1. $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \delta$, if $e_{ij} \in \mathcal{E}$,
2. $\|\mathbf{x}_i - \mathbf{x}_j\| > \delta$, if $e_{ij} \notin \mathcal{E}$.

Let $\mathbf{x}_i = (x_i, y_i)$ for all $1 \leq i \leq N$, then each of these relations can be written as inequality constraints, $\{f_k \geq 0\}$, where each $f_k \in \mathbb{R}[x_1, y_1, \ldots, x_N, y_N]$ is a polynomial in $2N$ variables over the real numbers. Therefore the realization problem is equivalent to asking if there exist $x_1, y_1, \ldots, x_N, y_N$ such that the following inequality constraints are satisfied.

$$\begin{aligned} \delta^2 - (x_i - x_j)^2 - (y_i - y_j)^2 &\geq 0, \text{ if } e_{ij} \in \mathcal{E}, \\ (x_i - x_j)^2 + (y_i - y_j)^2 - \delta^2 &> 0, \text{ if } e_{ij} \notin \mathcal{E}, \end{aligned}$$

where $1 \leq i < j \leq N$.

The following is an important result in studying problems of infeasibility for semi-algebraic sets [94].

Positivstellensatz: Given polynomials $\{f_1, \dots f_s\}$, $\{g_1, \dots, g_t\}$ and $\{h_1, \dots, h_u\}$, all in $\mathbb{R}[x_1, \dots x_n]$, the following are equivalent:

1. The set $\{x \in \mathbb{R}^n : f_i(x) \geq 0, h_i(x) = 0, g_i(x) \neq 0, i = 1 \dots s, j = 1 \dots t, k = 1 \dots u\}$ is empty.

2. There exist polynomials $f \in \mathcal{P}(f_1, \dots f_s), g \in \mathcal{M}(g_1, \dots g_t), h \in \mathcal{I}(h_1, \dots h_u)$ such that $f + g^2 + h = 0$,

where $\mathcal{P}(f_1, \dots f_s)$ is the *Cone* generated by the polynomials $\{f_i\}$, $\mathcal{M}(g_1, \dots g_t)$ is the *Monoid* over $\{g_i\}$ and $\mathcal{I}(h_1, \dots h_u)$ is the *Ideal* generated by $\{h_i\}$. See [94, 52] for further details on these algebraic objects.

4.2 Some Simplifications

Before using the Positivstellensatz, let us first perform the following simplifications. Note that if a formation $(\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N) \in \mathcal{C}^N(\mathbb{R}^2)$ is feasible, and $A : \mathbb{R}^2 \to \mathbb{R}^2$ is an isometry, then $(A\mathbf{x}_1, A\mathbf{x}_2, \cdots A\mathbf{x}_N)$ is also feasible. In fact, this induces an isometry of $C^N(\mathbb{R}^2)$ as well. Let us define an isometry A by an element of the group generated by translations and rotations in $\mathbb{R}^2$,

$$A(\mathbf{x}) = R(-\psi)(\mathbf{x} - \mathbf{x}_1),$$

where

$$R(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},$$

and ψ is the angle of the vector $\mathbf{x}_2 - \mathbf{x}_1$ w.r.t the positive x-axis in $\mathbb{R}^2$. With this choice, $A\mathbf{x}_1 = 0$ and $A\mathbf{x}_2 = [\|\mathbf{x}_2 - \mathbf{x}_1\|, 0]^T$. The existence of such an isometry implies that we can study the non-feasibility problem by ignoring the 3 variables x_1, y_1, and y_2. We therefore setup the quadratic forms in the following way. Let $M = 2N - 2$ and $\mathbf{x} = [x_2, x_3, y_3, x_4, y_4 \dots x_N, y_N, 1]^T \in \mathbb{R}^M$. If $e_{ij} \in \mathcal{E}$, then the inequality

$$\delta^2 - (x_i - x_j)^2 - (y_i - y_j)^2 \geq 0,$$

can be written as $\mathbf{x}^T A_{ij}\mathbf{x} \geq 0$, where $A_{ij} \in \mathbb{R}^{M\times M}$. If $j > i \geq 3$, then

$$A_{ij} = \left(\begin{array}{c|cc|c|cc|c|l} & & & & & & & \text{Row} \\ \hline & -1 & 0 & & 1 & 0 & & 2i-4 \\ & 0 & -1 & & 0 & 1 & & 2i-3 \\ \hline & & & & & & & \\ & & & & & & & \\ \hline & 1 & 0 & & -1 & 0 & & 2j-4 \\ & 0 & 1 & & 0 & -1 & & 2j-3 \\ \hline & & & & & & & \\ & & & & & & \delta^2 & 2N-2 \end{array}\right) \tag{8}$$

and all entries not explicitly written are zeros. It is easy to see that for terms involving $\mathbf{x}_1$ and $\mathbf{x}_2$, we have similar matrices with more zeros at the appropriate slots. In particular, if $e_{12} \in \mathcal{E}$ then the relation between $\mathbf{x}_1$ $\mathbf{x}_2$ simplifies to $x_2^2 \leq \delta^2$, with $A_{12}\text{diag}(-1, 0, \ldots, 0, \delta^2)$. Similarly, if $e_{lm} \notin \mathcal{E}$, then the inequality

$$(x_l - x_m)^2 + (y_l - y_m)^2 - \delta^2 > 0,$$

can be written as $\mathbf{x}^T B_{lm}\mathbf{x} > 0$. For $m > l \geq 3$ we have

$$B_{lm} = \left(\begin{array}{c|cc|c|cc|c|l} & & & & & & & \\ \hline & 1 & 0 & & -1 & 0 & & 2l-4 \\ & 0 & 1 & & 0 & -1 & & 2l-3 \\ \hline & & & & & & & \\ & & & & & & & \\ \hline & -1 & 0 & & 1 & 0 & & 2m-4 \\ & 0 & -1 & & 0 & 1 & & 2m-3 \\ \hline & & & & & & & \\ & & & & & & -\delta^2 & 2N-2 \end{array}\right)$$

We will have a total of $N(N-1)/2$ such quadratic forms. The non-feasibility problem is therefore equivalent to asking if the set $X = \{\mathbf{x} \in \mathbb{R}^M \mid \mathbf{x}^T A_{ij}\mathbf{x} \geq 0,\ 1 \leq i < j \leq N,\ e_{ij} \in$

$\mathcal{E}, \mathbf{x}^T B_{lm}\mathbf{x} > 0,\ 1 \leq l < m \leq N,\ e_{lm} \notin \mathcal{E}\}$ is empty. It is worthwhile to note that if the formation is feasible, the above simplification based on the isometry A, bounds the set X in the following way. We give the following standard definition.

The *diameter* D of a graph is defined as the longest graph geodesic between any two graph vertices of a graph.

$$D(G) = \max_{v_i, v_j \in \mathcal{V}} d(v_i, v_j),$$

where $d(v_i, v_j)$ is the graph geodesic between vertices v_i and v_j, given by the minimum length of the paths connecting them. In some sense, it is longest shortest path in a graph. Similar to this quantity, we can define the *graph radius* by,

$$\rho(G) = \max_{v_i \in \mathcal{V}} d(v_i, v_1),$$

We note that $\rho(G) \leq D(G)$. By placing $\mathbf{x}_1$ at $(0, 0)$, we therefore have $\|\mathbf{x}_i\| \leq \rho(G)\delta$ for $2 \leq i \leq N$. If $\mathbf{x} \in X$ then,

$$\mathbf{x}^T\mathbf{x} \leq (N-1)^2\delta^2\rho(G)^2 + 1. \tag{9}$$

Let $\mathcal{D} = \{\mathbf{x} \in \mathbb{R}^M \mid \mathbf{x}^T\mathbf{x} \leq (N-1)^2\delta^2\rho(G)^2 + 1\}$, then $X \subseteq \mathcal{D}$. Therefore for checking non-feasibility, we restrict our search to $\mathcal{D}$. Eq. (9) gives a useful bound on $\mathbf{x}$ and is particularly helpful for numerical computations. Moreover, it tells us that the search for a non-feasibility certificate is restricted to a bounded set. Unfortunately despite being bounded, X is not compact. We will see later that this complicates the search for the certificates.

4.3 Computations Using the $\mathcal{S}$-Procedure

It should be noted that all semi-algebraic constraints on the set X are quadratic. Moreover $A_{ij} = A_{ij}^T$ and $B_{lm}B_{lm}^T$. For quadratic constraints, the Positivstellensatz has been shown to be equivalent to the celebrated $\mathcal{S}$-procedure [87]. The $\mathcal{S}$-procedure transforms the Positivstellensatz into a linear matrix inequality (LMI) problem.

Theorem: Given symmetric $n \times n$ matrices $\{A_k\}_{k=0}^m$, the following are equivalent:

1. The set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T A_1\mathbf{x} \geq 0, \mathbf{x}^T A_2\mathbf{x} \geq 0, \cdots, \mathbf{x}^T A_m\mathbf{x} \geq 0, \mathbf{x}^T A_0\mathbf{x} \geq 0, \mathbf{x}^T A_0\mathbf{x} \neq 0\}$ is empty.

2. There exist non-negative scalars $\{\lambda_k\}_{k=1}^m$ such that $-A_0 - \sum_{k=1}^m \lambda_k A_k \geq 0$.

If $\{\lambda_k\}_{k=1}^m$ exist then let $\mathcal{Q} = -A_0 - \sum_{k1}^m \lambda_k A_k \geq 0$, $g = \mathbf{x}^T A_0 \mathbf{x}$ and

$$f = (\mathbf{x}^T \mathcal{Q} \mathbf{x})(\mathbf{x}^T A_0 \mathbf{x}) + \sum_{k=1}^{m} \lambda_k (\mathbf{x}^T A_0 \mathbf{x})(\mathbf{x}^T A_k \mathbf{x}).$$

Since $g \in \mathcal{M}(\mathbf{x}^T A_0 \mathbf{x})$ and $f \in \mathcal{P}(\mathbf{x}^T A_0 \mathbf{x}, \mathbf{x}^T A_1 \mathbf{x}, \cdots \mathbf{x}^T A_m \mathbf{x})$, $f + g^2 = 0$ as desired.

Note that any strict constraint $\mathbf{x}^T C \mathbf{x} > 0$ can be formulated as $\mathbf{x}^T C \mathbf{x} \geq 0, \mathbf{x}^T C \mathbf{x} \neq 0$. Therefore, if we ignore the strictness of all but one of the constraints given by $\mathbf{x}^T B_{lm} \mathbf{x} > 0$, the $\mathcal{S}$-procedure gives the certificates of non-feasibility for a given graph. Unfortunately, the strictness of more than inequality makes the $\mathcal{S}$-procedure *lossy* [17] i.e. it only becomes a sufficient condition. Therefore, this procedure can only work for the limited case described below.

4.4 Examples

Consider the complete graph $\mathcal{K}_N$. As discussed in Chapter 3, $\mathcal{K}_N$ is always feasible. It will be interesting to find if there is an example of an infeasible graph obtained by removing exactly one edge e, from K_N. We will denote such graph by $K_N - e$. Without loss of generality let $e = e_{12}$. By the procedure described above, the non-feasibility question can be answered by setting up $N(N-1)/2 - 1$ quadratic forms of the type $\mathbf{x}^T A_{ij} \mathbf{x} \geq 0$, $3 \leq i < j \leq N$ and one quadratic form of the type $\mathbf{x}^T B_{12} \mathbf{x} \geq 0$, where $B_{12} = \text{diag}(1, \cdots, -\delta^2)$. Clearly this can be a candidate for testing the $\mathcal{S}$-procedure described above. In fact, it can be seen that the corresponding LMI

$$\mathcal{Q} = -B_{12} - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \lambda_{ij} A_{ij} \tag{10}$$

can never be non-negative definite for any combination of $\lambda_{ij} > 0$. To prove this, note that $\mathcal{Q}$ can be written as

	x_2	x_3	y_3	x_4	y_4	.	x_i	y_i	.	x_N	y_N	1
x_2	a_{22}	a_{23}	0	a_{24}	.	.	a_{2i}	0	.	a_{2N}	0	0
x_3	a_{23}	a_{33}	0	a_{34}	0	.	a_{3i}	0	.	a_{3N}	0	0
y_3	0	0	a_{33}	0	a_{34}	.	.	a_{3i}	.	0	a_{3N}	0
x_4	a_{24}	a_{34}	0	a_{44}	0	.	a_{4i}	0	.	a_{4N}	0	0
y_4	0	0	a_{34}	0	a_{44}	.	0	a_{4i}	.	0	a_{4N}	0
⋮												⋮
x_i	a_{2i}	a_{3i}	0	a_{4i}	0	.	a_{ii}	0	.	a_{iN}	0	0
y_i	0	0	a_{3i}	0	a_{4i}	.	0	a_{ii}	.	.	a_{iN}	0
⋮	.	.	.	.	.	.	.	.	.	.	.	⋮
x_N	a_{2N}	a_{3N}	0	a_{4N}	0	.	a_{4i}	0	.	a_{NN}	0	0
y_N	0	0	a_{3N}	0	a_{4N}	.	0	a_{4i}	.	0	a_{NN}	0
1	0	0	0	0	0	.	0	0	.	0	0	a_{MM}

where the non-zero off-diagonal entries are

$$a_{mn} = -\lambda_{mn},\ m \neq n,$$

and the diagonal entries are given by

$$\begin{aligned}
a_{ii} &= -1 + \textstyle\sum_{j=3}^{N} \lambda_{2j}, && i = 2. \\
a_{ii} &= \lambda_{1i} + \textstyle\sum_{k=2}^{i-1} \lambda_{ki} + \sum_{k=i+1}^{N} \lambda_{ik}, && 2 < i \leq N. \\
a_{ii} &= \delta^2(1 - \textstyle\sum_{k=1}^{N-1} \sum_{j=k+1}^{N} \lambda_{kj}) && i = M.
\end{aligned}$$

Notice that the feasibility question is independent of δ. If G is feasible for $\delta_1 > 0$ then it is feasible for all $\delta > 0$. For simplicity let $\delta = 1$. Since $\lambda_{ij} > 0$, $\mathcal{Q} \neq 0$. So we need to show that $\mathcal{Q} \ngtr 0$. Note that this is *not* equivalent to showing that $\mathcal{Q} < 0$. A necessary condition for $\mathcal{Q} = [q_{ij}] > 0$ is that

$$q_{ii} > 0$$

$$q_{ii} + q_{jj} > 2q_{ij}$$

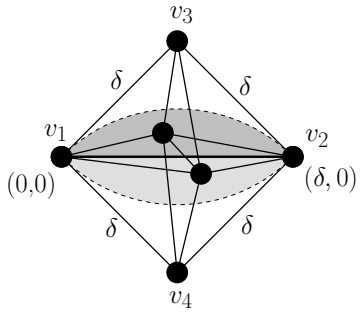


Figure 7: Geometric construction of a feasible formation for $K_N - e$.

Now consider,

$$\begin{aligned} a_{22} + a_{MM} &= -1 + \sum_{j=3}^{N} \lambda_{2j} + 1 - \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} \lambda_{ik} \\ &= -\sum_{j=1}^{N} \lambda_{2j} - \sum_{i=3}^{N-1} \sum_{k=i+1}^{N} \lambda_{ik} \end{aligned}$$

If $\mathcal{Q} > 0$ then $a_{22} + a_{MM} > 0$, but $\lambda_{ij} > 0$ for all $1 \leq i < j \leq N$, so that $a_{22} + a_{MM} < 0$, which is a contradiction. So $\mathcal{Q}$ is never non-negative definite for any choice of λ_{ij}. By the $\mathcal{S}$-procedure, the non-negativity condition of $\mathcal{Q}$ is both necessary and sufficient. Therefore, the set is always non-empty and we have proved the following result.

Proposition: $\mathcal{K}_N - e$ is feasible for all N.

Motivated by this result, one can also give a geometric construction to provide feasible formations for $\mathcal{K}_N - e$ for all N. Consider the realization of the graph depicted in the lower left corner of Figure 5. This is a realization $\mathcal{K}_4 - e$. Now consider the Figure7. The shaded region is the intersection of discs of radius δ, centered at x_3 and x_4.

For the general case, where we have multiple strict inequalities, we present the following result for odd number of strict constraints.

Proposition: Let $Q \in \mathbb{N}$ be odd. Given symmetric $n \times n$ matrices $\{A_k\}_{k=0}^{P}$, the set $Y = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T A_0 \mathbf{x} \geq 0, \cdots, \mathbf{x}^T A_{Q-1} \mathbf{x} \geq 0, \mathbf{x}^T A_Q \mathbf{x} \geq 0, \cdots, \mathbf{x}^T A_P \mathbf{x} \geq 0, \mathbf{x}^T A_0 \mathbf{x} \neq 0, \cdots, \mathbf{x}^T A_{Q-1} \mathbf{x} \neq 0, \}$ is empty if for all $0 \leq q \leq Q-1$, there exist non-negative scalars $\{\lambda_k^{(q)}\}$, $1 \leq k \leq P, k \neq q$ such that

$$-A_q - \sum_{k=1, k \neq q}^{P} \lambda_k^{(q)} A_k \geq 0.$$

Proof: Let $\mathcal{Q}_q = -A_q - \sum_{k=1,k\neq q}^{P} \lambda_k^{(q)} A_k$. If $\mathcal{Q}_q \geq 0$ then let

$$g(\mathbf{x}) \triangleq \prod_{q=0}^{Q-1} (\mathbf{x}^T A_q \mathbf{x}) \in \mathcal{M}(\mathbf{x}^T A_0, \cdots, \mathbf{x}^T A_{Q-1}\mathbf{x}),$$

$$\begin{aligned} f(\mathbf{x}) \quad &\triangleq \quad \prod_{q=0}^{Q-1} (\mathbf{x}^T A_q \mathbf{x})(\mathbf{x}^T \mathcal{Q}_q \mathbf{x}) + \cdots \\ \cdots &+ \sum_{i_1,\cdots i_{Q-1}} \lambda_{j_1}^{i_1} \cdots \lambda_{j_{Q-1}}^{i_{Q-1}} (x^T A_{i_1} x) \ldots (x^T A_{i_{Q-1}} x) \end{aligned}$$

Here the details on the indexes $i_1, \ldots, i_{Q-1}$ and $j_1, \ldots, j_{Q-1}$ have been omitted for brevity. Clearly, $f \in \mathcal{P}(\mathbf{x}^T A_0, \cdots, \mathbf{x}^T A_P \mathbf{x})$ so that $f + g^2 = 0$ as desired. ∎

4.5 Open Problem I: Necessary Conditions for Infeasibility

Note that the results presented in this chapter only provide sufficient conditions for the non-feasibility of connectivity graphs. Finding a necessary criterion is still an open problem. With such a criterion, it will be possible to give a complete geometric characterization of the entire class of connectivity graphs, significantly improving the results presents here.

CHAPTER V

STRUCTURAL COMPLEXITY OF FORMATIONS

When designing control strategies for distributed, multi-agent robot systems, it is vitally important that the number of prescribed local interactions is managed in a scalable manner. In other words, it should be possible to add new robots to the system without causing a significant increase in the communication and computational burdens of the individual robots. But, one strategy that achieves this is a strategy where no interactions are present, which is clearly unsatisfactory from a number of vantage points. Therefore, an additional requirement when designing multi-agent coordination strategies should be that enough local interactions are present in order to ensure the proper execution of the task at hand.

Hence, a fundamental question that arises when studying such multi-agent systems is how to properly define the notion of "complexity". The traditional, algorithmic notion of the complexity of a system is related to how difficult it is to *describe* it. Therefore, most of the measures of complexity are closely related to the *Algorithmic Information Content* (AIC) in a system [6]. However, as noted in the physical sciences [14, 16, 88], there is an inherent difference between *descriptive complexity* and *structural complexity*, where the latter measures the interactions, size, and asymmetry in the physical structure. As of yet, no universal concept of structural complexity has emerged. As an example, molecular chemists define the complexity of molecules relative to the particular problem under investigation [88]. A similar program can be carried out within the context of formation control. It is clear that when talking about robot formations, any measure of the complexity of the formations should take into account the size of the formation, the number of communication links or interactions in the formation, and possibly also the degree of symmetry in the formation.

Molecular chemists have mainly described the structural complexity of molecules by defining measures on their corresponding graphs [14]. Since, there is a corresponding notion of formation graphs induced by robot formations, as described in the previous chapters, the

structural information in the formation is indeed captured by these graphs. Therefore, it seems appropriate to study the structural complexity of multi-agent robot formations using connectivity graphs.

5.1 Descriptive Vs Structural Complexity

It should be emphasized that if we were to use descriptive measures of complexity, like the AIC, a formation that has a complete connectivity graph, e.g. K_5, as shown in Figure 8, would be easier to describe than the "ring-like formation", with corresponding connectivity graph C_5. However, this is counter intuitive to our notion of complexity in robot formations since there are more *local interactions* in K_5 than in C_5, and therefore K_5 is expected to have greater structural complexity. A less straight forward situation is illustrated in Figure 9. Should less interactions necessarily mean less complexity in this case? One may argue that C_5 should be less complex because it has more symmetry than the star graph S_4, despite having one more link. It can moreover be noted that star graphs cease to exist as connectivity graphs above $N = 6$, as shown in Chapter 3. It should be noted, however, that when formulating a measure of complexity for robot formations, it need not produce an absolute order on all connectivity graphs (although the order has to be observed in its own class e.g. among all rings, all stars, all complete graphs). This means that we are more interested in relative complexity. We should thus be able to at least differentiate between *very complex* formations and *very simple* ones. Hence, the complexity measure need not be unique for every graph.

There is a distinction made in molecular chemistry between *intrinsic* and *extrinsic* complexities [14]. When studying the complexity of a molecule according to the number of interactions in the structure (bonds), symmetry does not play a role since symmetric structures do not necessarily imply fewer interactions. This is referred to as *intrinsic* complexity. But, when studying problems of synthesis or fabrication, in which a process is iterated to build a complex structure, symmetry clearly implies lesser complexity [88]. The same approach should be taken when studying robot formations. The complexity of a robot formation that is already in place should not be affected by the potential symmetries of its

Figure 8: C_5 vs K_5

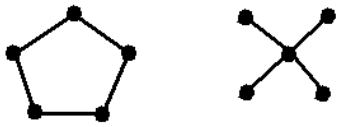


Figure 9: C_5 vs S_4

corresponding connectivity graph. However, when studying the problem of *synthesizing* formations, it can be argued that it is easier to produce symmetric structures [58]. This is the case since in a completely symmetric formation, the same program can be executed by the different robots, which is not the case for asymmetric formations. However, in this work we limit the scope to the intrinsic complexities, and leave extrinsic complexities to a future endeavor.

5.2 Perception, Communication, and Information Flow

Individual agents in a formation can interact with each other in two distinct ways, via perception or communication. However, both of these two types of interactions result in a total information flow in the system, which means that they should both be taken into account when the complexity measure is defined. The sensors are used to gain information about other agents and the environment, while the communication channels are used to directly relay information to other agents. The concept of information flow among agents should thus be tied to the complexity of the formation in order to capture interactions in terms of perception and communications respectively. However, how does information flow due to perception differ from information flow due to communication? Are these two sides of the same coin or should they be treated as fundamentally different?

Given the above mentioned considerations, we will define a complexity measure of robot formations, related to the complexity of its connectivity graphs, which has a remarkable similarity with the complexity measures defined on graphs by molecular chemists. In [88], the complexity of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has been suggested as

$$C(G) = \sum_{v_i \in \mathcal{V}} \left(deg(v_i) + \sum_{v_j \in V, v_i \neq v_j} \frac{deg(v_j)}{d(v_i, v_j)} \right), \tag{11}$$

where $d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ is some distance function defined between vertices. As we will see, this definition will help us to characterize intrinsic complexity of robot formations quite nicely.

In the following pages, we will first discuss the equivalence between perception and communication from an information theoretic point of view. Then, we will propose a definition of the intrinsic complexity of robot formations, and explain its relation to the complexity of graphs.

5.3 Perception Vs. Communication

Any measure of how complex a certain formation is has to capture the amount of information that flows between the different agents in a meaningful manner. This exchange of information between agents is due to the local interactions among agents. There are two kinds of local interactions in multi-agent robotic systems. One is due to sensory perception of neighboring robots and the other is due to the communication channels. When defining complexity measures, one thus either have to unify these two types of local interactions, or define two different complexity costs associated with them. Hence, it is natural to ask whether these interactions differ fundamentally from each other. If we can show that there is no fundamental difference, it will simplify our task of characterizing complexity in terms of local interactions by not explicitly mentioning the cause of the interactions. We explore this issue in this section.

Since we are interested in this issue from an information theoretic point of view, we pose this problem in an information theoretic setting. If X, Y are two random variables, we will denote by $I(X;Y)$, the amount of information gained about X by knowing Y. The

entropy of each random variable will be denoted by $H(X)$ and $H(Y)$ respectively, and $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ (see for example [24, 93]), where $X|Y$ and $Y|X$ are conditional random variables. A variable Z of M components can be defined over a quantization lattice $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \cdots \times \mathbb{Z}_{k_M} \subset \mathbb{R}^M$, where $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$. Such a variable can be used to model the quantization effects when sampling a physical quantity by a sensor of finite resolution, or the loss of information when encoding some quantity for transmission over a bandlimited channel.

Problem 5.3.1 *Suppose the state of a system $X = [x_1, x_2, \dots x_M]^T \in \mathbb{R}^M$ is measured by a sensor $\mathcal{S}$, providing the measurements $Z[z_1, z_2 \dots z_M]^T \in \mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \cdots \times \mathbb{Z}_{k_M} \subset \mathbb{R}^M$, where $k_i \in \mathbb{N}$ for $1 \leq k \leq M$. Knowledge about X is also transmitted by a remote agent over a communication channel $\mathcal{C}$ as a vector $Y = [y_1, y_2, \dots y_M]^T \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \cdots \times \mathbb{Z}_{N_M} \subset \mathbb{R}^N$, where $N_i \in \mathbb{N}$ for $1 \leq i \leq N$. Here, the component y_i of Y is a (possibly incomplete) description of the component x_i of X. Each component y_i is encoded independently of other components, and each symbol in each component is equally likely. i.e. $p_i(y_i) = \frac{1}{N_i}$. Then, we ask the following questions:*

1. *Does there always exist a virtual sensor S' which provides the same information as the communication channel $\mathcal{C}$?*
2. *Does there always exist a virtual communication channel $\mathcal{C}'$ which provides the same information as the sensor $\mathcal{S}$?*

These two questions seem to be the same in the sense that they both ask for sufficient conditions for the following to hold:

$$I(X;Z) = I(X;Y) \quad \text{or}$$
$$H(X|Z) = H(X|Y),$$

but we will see, in what follows, that Question 1 is easier to answer.

5.3.1 Example

We start our investigation by studying a simplified, yet illustrative example: Suppose that robot i is transmitting its position to robot j over a lossless communication channel. It

is moreover assumed that robot i measures its position with respect to $(0,0)$, i.e. with respect to robot j. It encodes its radial position r as $\tilde{r}$ using N_r symbols $\{r_1, r_2 \dots r_{N_r}\}$, with resolution $\frac{\delta}{N_r}$, using the probability distribution $p(\tilde{r})$. Similarly, if there are N_θ symbo ls $\{\theta_1, \theta_2 \dots \theta_{N_r}\}$ for quantizing the bearing $\tilde{\theta}$, with resolution $\frac{2\pi}{N_\theta}$, using the distribution $p(\tilde{\theta})$, then the joint probability distribution of the symbol set is $p(\tilde{r}, \tilde{\theta})$. The entropy of the coding scheme is given by

$$\begin{aligned} H(\tilde{r}, \tilde{\theta}) &= -\sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} p(r_i, \theta_j) \log_2 p(r_i, \theta_j) \\ &= H(\tilde{r}) + H(\theta|\tilde{r}). \end{aligned}$$

We now assume that $\tilde{r}$ is independent of $\tilde{\theta}$, so that $p(\tilde{r}, \tilde{\theta}) = p(\tilde{r})p(\tilde{\theta})$ and $p(\tilde{\theta}|\tilde{r})p(\tilde{\theta})$. Furthermore, assume that all symbols are equally likely i.e. $p(\tilde{r}) = \frac{1}{N_r}$ and $p(\tilde{\theta}) = \frac{1}{N_\theta}$. We then get

$$\begin{aligned} H(\tilde{r}, \tilde{\theta}) &= H(\tilde{r}) + H(\tilde{\theta}|\tilde{r}) = H(\tilde{r}) + H(\tilde{\theta}) \\ H(\tilde{r}, \tilde{\theta}) &= \log_2(N_\theta + N_r) \end{aligned}$$

The assumptions made here about uniform resolutions and independence are reasonable as they are usually true for real sensors. It is the processing (e.g. polar to Cartesian conversion in this case), that produces non-uniform resolutions or correlations.

5.3.1.1 Communication

Suppose that position $(\tilde{r}, \tilde{\theta})$ is transmitted to robot j from robot i. Then the amount of information that is gained about robot i's true position, (r, θ), is given by the mutual information

$$I(r, \theta; \tilde{r}, \tilde{\theta}) = H(r, \theta) - H(r, \theta|\tilde{r}, \tilde{\theta}).$$

Since robot i is equally likely to be in any position in $\mathcal{B}_\delta(0)$, we may assume that

$$f(r, \theta) = f(r)f(\theta) = \frac{1}{\delta}\frac{1}{2\pi} = \frac{1}{2\pi\delta}.$$

Now, the uncertainty associated with robot i's true position can thus be calculated as

$$\begin{aligned} H(r,\theta) &= H(r) + H(\theta|r) = H(r) + H(\theta) \\ H(r,\theta) &= -\int_0^{\delta} f(r)\log_2 f(r)dr - \int_0^{2\pi} f(\theta)\log_2 f(\theta)d\theta \\ &= \log_2\delta + \log_2(2\pi) = \log_2(2\pi\delta). \end{aligned}$$

Next we compute $H(r,\theta|\tilde{r},\tilde{\theta})$. Observe that:

$$\begin{aligned} H(r,\theta|\tilde{r},\tilde{\theta}) &= H(r|\tilde{r},\tilde{\theta}) + H(\theta|r,\tilde{r},\tilde{\theta})) \\ &= H(r|\tilde{r}) + H(\theta|\tilde{\theta}), \end{aligned} \tag{12}$$

where

$$\begin{aligned} H(r|\tilde{r}) &= \sum_{i=1}^{N_r} p(r_i)\left(\int f(r|r_i)\log_2 f(r|r_i)dr\right) \\ H(\theta|\tilde{\theta}) &= \sum_{i=1}^{N_\theta} p(\theta_i)\left(\int f(\theta|\theta_i)\log_2 f(\theta|\theta_i)d\theta\right). \end{aligned}$$

If we now assume a uniform quantizer, we get the uniform distributions:

$$\begin{aligned} f(r|r_k) &= \frac{N_r}{\delta},\ r \in [r_k - \frac{\delta}{2N_r}, r_k + \frac{\delta}{2N_r}],\ 1 \le k \le N_r \\ f(\theta|\theta_k) &= \frac{N_\theta}{2\pi},\ \theta \in [\theta_k - \frac{\pi}{N_\theta}, \theta_k + \frac{\pi}{2N_\theta}],\ 1 \le k \le N_\theta. \end{aligned}$$

Therefore,

$$\begin{aligned} H(r|\tilde{r}) &= -\sum_{i=1}^{N_r} \frac{1}{N_r}\left(\int_{r_k - \frac{\delta}{2N_r}}^{r_k + \frac{\delta}{2N_r}} \frac{N_r}{\delta}\log_2\frac{N_r}{\delta}dr\right) \\ H(r|\tilde{r}) &= \log_2\frac{\delta}{N_r}. \end{aligned}$$

Similarly,

$$\begin{aligned} H(\theta|\tilde{\theta}) &= \log_2\frac{2\pi}{N_\theta} \\ \text{So } H(r,\theta|\tilde{r},\tilde{\theta}) &= \log_2\left(\frac{2\pi\delta}{N_r N_\theta}\right). \end{aligned}$$

The amount of information gained by communication is therefore

$$\begin{aligned} I_c(r,\theta;\tilde{r},\tilde{\theta}) &= H(r,\theta) - H(r,\theta|\tilde{r},\tilde{\theta}) \\ &= \log_2 N_r N_\theta. \end{aligned}$$

5.3.1.2 Perception

Now, consider the situation where robot j is sensing robot i's true position by an on-board sensor with radial resolution Δr and angular resolution $\Delta\theta$.[1] Moreover, assume that the range of the sensor is limited to R i.e. $\hat{r} \leq R$. The sensor then gives the position $(\hat{r},\hat{\theta})$. By repeating this argument in the angular case, we get

$$H(r,\theta|\hat{r},\hat{\theta}) = H(r|\hat{r},\hat{\theta}) + H(\theta|r,\hat{r},\hat{\theta})).$$

If we further assume that the sensor measurements $\hat{r}$ and $\hat{\theta}$ are mutually independent, then

$$H(r,\theta|\hat{r},\hat{\theta}) = H(r|\hat{r}) + H(\theta|\hat{\theta}),$$

where

$$\begin{aligned} H(r|\hat{r}) &= -\sum_{i=0}^{R/\Delta r-1} p(i\Delta r)\int_0^R f(r|i\Delta r)\log_2 f(r|i\Delta r)dr \\ H(\theta|\hat{\theta}) &= -\sum_{i=0}^{2\pi/\Delta\theta-1} p(i\Delta\theta)\int_0^{2\pi} f(\theta|i\Delta\theta)\log_2 f(\theta|i\Delta\theta)d\theta. \end{aligned}$$

5.3.1.3 Equivalent Virtual Sensors

Let us now try to build an equivalent virtual sensor to the communication channel described above. Suppose there exists a sensor with independent measurements $\hat{r},\hat{\theta}$, whose range is limited to $R = \delta$. Let the resolutions of the sensor measurements be $\Delta r = R/N_r$,

[1] This is a reasonable assumption, since most range sensors such as ultrasonic or infra-red sensors, are typically mounted in a ring of individual sensors with a certain, fixed range resolution [63].

$\Delta\theta = 2\pi/N_\theta$ respectively. If the errors $e_r = r - \hat{r}$, $e_\theta = \theta - \hat{\theta}$ in the sensor's measurements are modeled by uniform distributions, given by:

$$f(e_r) = \frac{1}{\Delta r}, \qquad e_r \in [-\frac{\Delta r}{2}, \frac{\Delta r}{2}],$$
$$f(e_\theta) = \frac{1}{\Delta \theta}, \qquad e_\theta \in [-\frac{\Delta \theta}{2}, \frac{\Delta \theta}{2}],$$

then it can be seen directly that $I_c(r, \theta; \hat{r}, \hat{\theta}) = I_p(r, \theta; \tilde{r}, \tilde{\theta}) = \log_2(N_r N_\theta)$.

This thus provides us with sufficient conditions for the equality to hold between perception and communications.

5.3.2 The General Case

Theorem 5.3.1 *For any communication link $\mathcal{C}$ that satisfies the assumptions in Problem 5.3.1, there always exist a virtual sensor S' that provides the same information as the communication channel.*

Proof:

By the setup in Problem 5.3.1, we have

$$I(X;Y) = \log_2(\prod_{i=1}^{N} N_i).$$

If we follow steps, similar to the previous example, we can construct a "virtual sensor" $\mathcal{S}'$ which is equivalent to the communication channel $\mathcal{C}$ as follows. Let the virtual sensor give measurements $Z' \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \cdots \times \mathbb{Z}_{N_M} \subset \mathbb{R}^M$, with resolutions

$$\Delta z_i' = \frac{\max(y_i) - \min(y_i)}{N_i}$$
$$f(z_i' - x_i) = f(x_i | i\Delta z') = \frac{1}{\Delta z_i}$$

Then it can be directly verified that

$$I(X;Z') = H(X;Y) = \log_2(\prod_{i=1}^{N} N_i).$$

■

Now, we come to our second question, namely the problem of creating a "virtual" communication channel $\mathcal{C}'$ equivalent to a given sensor. If $I(X;Z)$ is the amount of information

gained about X by measurement Z, and there exists a positive integer k such that

$$k = 2^{I(X;Y)} \in \mathbb{Z}^+,$$

then we can build a virtual communication channel $\mathcal{C}'$ using any factorization of k

$$k = k_1.k_2.\ldots.k_K, \qquad k_i \in \mathbb{Z}^+.$$

However, it is usually the case that $I(X;Y) \in \mathbb{R} \setminus \mathbb{Z}^+$ due to the choice of continuous, non-constant distributions, defined over intervals in the field $\mathbb{R}$. Therefore it may not always to be possible to construct the virtual channel, using this "trick". But, by the aforementioned argument, we assume that we can talk about sensors and communications channels interchangeably.

5.4 Complexity of Robot Formations

We now consider the problem of describing a complexity measure of multi-agent robot formations. As explained above, it would be appropriate to relate this measure to the number of local interactions between agents. In the previous section, we saw that these interactions are due to perception and communication, and that there is no fundamental difference between the two, from an information theoretic point of view. Therefore it makes sense to relate the complexity measure to the total amount of information flowing in the system. It should further be noted that this information exchange among agents is a dynamic quantity and depends on the distributed algorithm executed by the system.

A multi-agent formation is an evolving structure in both time and space. In space, it is dynamic due to the motion of the robots, which leads to the establishment of new interactions and the termination of old ones. This spatial relationship can be captured by a connectivity graph as explained in Chapter 3. However, the establishment of a local interaction, does not mean that this interaction is present for all time. The information exchange at a particular time depends on *protocols* (e.g. [57, 50]), which may make the information interchange not only non-constant, but also non-deterministic. Therefore, it would be appropriate to refer to a quantity describing the time rate of information exchange.

We call this quantity, the *information flow*, and refer to the complexity of a formation as the total information flow in the system.

In [57], complexity measures for multi-agent, distributed algorithms were given. Moreover, explicit scaling laws were proved with respect to the number of agents. However, the implicit assumption in that work is that all agents can communicate with all other agents directly. Therefore, the results are limited to complete graphs only, and can be said to measure the complexity of the communication strategies rather than the complexity of the formations. To remedy this, we provide an account of the structural complexity of formations that is applicable to formations in which agents have limited information about other agents.

5.4.1 States, Measurements, and Packets

It would be appropriate to say something about the nature of the information flowing between agents. In multi-agent coordination strategies, the states that are being shared need not be physical states of a dynamical system associated with an agent or the environment. The states, about which a remote agent gains knowledge, via perception or communication, may moreover be discrete or continuous. However, the outputs of both sensors and communication channels are discrete quantities. If a discrete quantity $X \in \mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \cdots \times \mathbb{Z}_{k_N}$, is coded and transmitted as Z, without modification over a communication channel (or acquired by perception without errors), and decoded reversibly, then there is no uncertainty associated with the communication (or measurement) and it describes the true state in its entirety. In this case $I(X;Z) = H(X) = H(Z)$. However if the state is defined over a continuous field, or the coding and transmission process is not reversible, then there is a loss of information in the perception and communication processes, as $I(X;Z) = H(X) - H(X|Z)$.

Let us now suppose that an agent receives a sequence of measurements (or packets) about an evolving state $\{X(1), X(2), \ldots X(n)\}$ as $\{Z(1), Z(2), \ldots Z(n)\}$. The amount of information gained at time n is $I(X; Z(n), Z(n-1), \ldots Z(1))$. This is because it may be possible to predict the true state using all previous observations, using an estimator. However, if the random process $X(n)$ is white, i.e. previous values have no correlation with

the latest one, then the information gained is $I(X(n); Z(n))$. If the process is Markovian, then $I(X(n); Z(n), Z(n-1))$, and so on [24]. Therefore, we will use $I(X; Z)$ to denote the information exchange, even though this quantity may actually be dependent on previous observations. Also without loss of generality, we will refer to the flow of continuous states $X \in \mathbb{R}^N$, only, but would not mean that only continuous quantities are being quantized and exchanged in the system.

5.4.2 Protocols, Synchronous Algorithms, and Information Flows

Suppose $X_j \in \mathbb{R}^N$ is a state associated with an agent j, which agent i wants to acquire by perception or communication. Let $Z_{j,i} \in \mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \cdots \times \mathbb{Z}_{k_N} \subset \mathbb{R}^N$ be the measurement of a sensor $\mathcal{S}$ by agent i. Information about X_j is also transmitted by agent j over a communication channel $\mathcal{C}$ as $Y_{j,i} \in \mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \cdots \times \mathbb{Z}_{p_N} \subset \mathbb{R}^N$, where $p_i \in \mathbb{N}$ for $1 \leq i \leq N$. If we consider $X_j, Z_{j,i}$ and $Y_{j,i}$ as random processes, then we can define the *information flow*, as the time rate of information exchange taking place at a certain agent, i.e.

$$F_{i,j}(t) = \frac{dI(X_j; Z_{j,i}, Y_{j,i})}{dt}. \tag{13}$$

There are several technical difficulties associated with the definition in Equation (13). The random processes are always discrete in time, because both the perception and communication process are discrete. In the most general case, the packets arrive (or measurements are taken) according to some *protocol*, which defines the time of arrival. The situation is further complicated by the fact that the information exchange may be completely asynchronous, both among different agents as well as between measurements and communication of the same state for one agent. The actual communication exchange takes place as a burst after possibly long unequal intervals. But, in this paper, we assume that the information flow for a single exchange should be considered as the information gained between two consecutive exchanges, averaged over the time interval.

With these considerations, we assume that if the information flow is well defined according to a particular protocol, then we can define the intrinsic structural complexity of a formation as follows.

Definition 5.4.1 (Intrinsic Complexity of a Formation) *The intrinsic complexity of a formation* $\mathcal{F} = (X_1, X_2, \ldots X_N) \in \mathcal{C}^N(\mathbb{R}^2)$ *is defined as:*

$$C(\mathcal{F}) = \sum_j \sum_{i \neq j} F_{i,j}(X_j),$$

where each $F_{i,j}$ *is defined according to some given communication protocol.*

The asynchrony in information exchange, and the presence of a particular protocol, lead to another interesting phenomenon. Intuitively, it seems reasonable to assume that the total complexity of the system is the sum of all local interactions. However, due to the presence of protocols and asynchrony, this complexity varies over time as the time averages may vary. It is easy to see that this variation is not contributed as much by asynchrony but by complex protocols of information exchange. We therefore assume here for simplicity that the multi-agent algorithm executed is synchronous, and leave the issue of asynchronous algorithms for future investigation. We will see below that this is a reasonable assumption to start with, and the results obtained are very useful as a first analysis of structural complexity. However the presence of protocols is not something that can be overlooked without a justification.

Since, the presence of protocols implies that every interaction is not active during a certain time period, the intrinsic complexity is bounded above by a quantity that assumes that all interactions are active for all time. This bound is in-fact a complexity associated with a *broadcast protocol*, defined below.

Definition 5.4.2 (Synchronous Periodic Broadcast Protocol) *Suppose each agent* j *transmits its state* $X_j, j \neq i$ *to all other agents as* Y_j *after every* Δt *seconds. The time* Y_j *takes to reach agent* i *is some integer multiple* $k_{i,j}$ *of* Δt*, where* $k_{i,j}$ *is the number of "hops" in the communication. Also, let the measurement* $Z_{j,i}$ *of remote state* X *be periodically taken every* Δt *seconds. Then this protocol of communication among agents is called the* Synchronous Periodic Broadcast Protocol.

If Δt is the minimum permissible time for information exchange in the system (due to either bandwidth, sensor update interval, or algorithm execution cycle), then we can easily see that protocols of synchronous information exchange that are more complicated than the

broadcast protocol would result in a decrease of the total information flow. If we denote the complexity of a formation, associated with the broadcast protocol as $C_B(\mathcal{F})$, then

$$C_B(\mathcal{F}) \geq C_P(\mathcal{F}),$$

where $C_P(\mathcal{F})$ is the complexity for some arbitrary protocol. $C_B(\mathcal{F})$ is therefore the worst case complexity associated with a particular formation. The information flow of a remote state X_j at agent i, according to this protocol, is

$$F_{i,j}(X_j) = \frac{I(X_j; Z_{j,i})}{\Delta t} + \frac{I(X_j; Y_j)}{k_{i,j}\Delta t} \text{ bits/sec}, i \neq j.$$

From the discussion in Section 5.3, it is clear that it is always possible to create a virtual sensor $\mathcal{S}'$ such that $I(X_j; Y_j) = I(X_j; {Z_j}')$. Therefore, we will refer to the information flows with reference to sensors only, and write the information flow as

$$F_{i,j}(X_j) = \frac{I(X_j; \mathcal{Z}_{j,i})}{k_{ij}\Delta t}, \tag{14}$$

where $\mathcal{Z}_{j,i} = [Z_{j,i}, Z'_j]$, in order to emphasize that we are referring to sensors only.

5.4.3 Complexity and Connectivity Graphs

We now study the interesting relationship between the structural complexity defined above and an alternative description of complexity based on connectivity graphs of formations. The first interesting connection can be seen from the definition of the broadcast protocol. The number k_{ij} defined as the number of hops in the communication between agents hints at the network topology between the agents. But, the connectivity graphs defined in Chapter 3 are exactly these network topologies. Furthermore, it may be reasonable to ask if k_{ij} is a unique number for any two agents, since the same information my be exchanged by different hopping paths. This corresponds to different paths in the connectivity graph. Since the information flow in Equation 14 depends on k_{ij}, it must be made clear what path are we using. But, since we are interested in distributed multi-agent algorithms, it cannot be assumed that global information about the network topology (i.e. the connectivity graph of the formation) is available all the time to all agents, so that the hopping paths are unique[2].

[2]Network discovery may be possible eventually, but not guaranteed for all time, specially right after the formation graph switches to a new one.

Instead, in the broadcast scenario, the information about X_j reaches a remote agent i via all possible hopping paths between them, so that

$$F_{i,j}(X_j) = \sum_{p=1}^{P_{ij}} \frac{I(X_j; \mathcal{Z}_{j,i})}{k_{p,ij}\Delta t}, \tag{15}$$

where P_{ij} is the total number of paths, and $k_{p,ij}$ is the length of an individual path, p. If k_{ij} is the smallest path between the agents, i.e. a geodesic in the corresponding connectivity graph, then

$$F_{i,j}(X_j) \leq deg(v_j)\frac{I(X_j; \mathcal{Z}_{j,i})}{k_{ij}\Delta t}. \tag{16}$$

Furthermore the complexity $C_B(\mathcal{F})$ is bounded above as

$$C_B(\mathcal{F}) \leq \sum_j \sum_{i \neq j} deg(v_j)\frac{I(X_j; \mathcal{Z}_{j,i})}{k_{ij}\Delta t}.$$

We now assume that the states exchanged by all agents are of the same type and encoded in the same way. Therefore $I(X_j; \mathcal{Z}_{i,j}) = \gamma$, i.e. the mutual information is constant for all i, j. Also, note that $k_{ij} = 1$ if v_i, v_j make an edge in the connectivity graph i.e. when agent j can be directly sensed (or communicated with) without an additional hop. We can also write this in standard graph theory notation as $v_j \in star(v_i)$ [29, 47]. Using this notation, we have:

$$C_B(\mathcal{F}) \leq \frac{\gamma}{\Delta t}\sum_i \left(\sum_{v_j \in star(v_i)} deg(v_j) + \sum_{v_j \notin star(v_i)} \frac{deg(v_j)}{k_{ij}} \right).$$

It should further be noted that if $v_j \in star(v_i)$, the exact path of communication is *always* known, and the broadcast to other nodes is not necessary. Therefore we can make this bound tighter

$$C_B(\mathcal{F}) \leq \frac{\gamma}{\Delta t}\sum_i \left(deg(v_i) + \sum_{v_j \notin star(v_i)} \frac{deg(v_j)}{k_{ij}} \right),$$

where $\sum_{v_j \in star(v_i)} 1 = deg(v_i)$. Compare this to the complexity defined on a graph, in the context of molecular chemistry, given in Equation 11, and we get,

$$C_B(\mathcal{F}) \leq \frac{\gamma}{\Delta t} C(\Phi_N(\mathcal{F})),$$

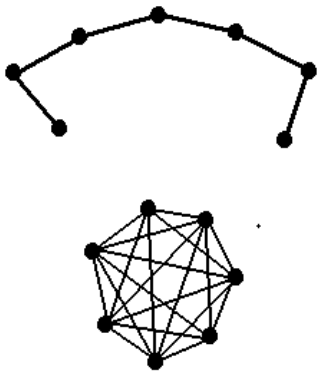


Figure 10: δ-chain and complete graph for 7 vertices.

where $\Phi_N(\mathcal{F})$ is the connectivity graph of the formation. This relationship leads to the following interesting observation:

The complexity of the connectivity graph of a formation is a (tight) upper bound for the worst case complexity associated with an arbitrary protocol of communication in a multi-agent formation.

Therefore the study of structural complexity of robot formations is closely related to the complexity of their connectivity graphs.

5.4.4 Simple and Complex Connectivity Graphs

The complexity measure on connectivity graphs gives a good comparison between different formations. While it is difficult to produce an absolute order on all possible connectivity graphs, it distinguishes simple graphs from the more complex. We will prove below that the complete graph is the most complex connectivity graph for a fixed set of vertices, whereas a δ-chain, which is a Hamiltonian path on all vertices, is the least complex connected connectivity graph. (See Figure 10.)

The conclusion that the complete graph is the most complex graph is not surprising and confirms to our intuition, as it has the maximum number of local interactions between any set of vertices. The characterization of the most simple graph is however an interesting result and gives the justification of the δ-chaining algorithms, as described in Chapter 12.

Consider a connectivity graph $G_N = (\mathcal{V}, \mathcal{E})$ on N vertices, with the complexity measure

$$C(G_N) = \sum_{v_i \in \mathcal{V}} \left(deg(v_i) + \sum_{v_j \notin star(v_i)} \frac{deg(v_j)}{k_{ij}} \right).$$

If we add another vertex v_{N+1} to G_N, we get a graph on $N+1$ vertices G_{N+1}. We can also form new edges between v_{N+1} and vertices in V so that the complexity of the new graph is perturbed as

$$\begin{aligned} C(G_{N+1}) = & \sum_{v_i \in \mathcal{V}} (deg(v_i) + \Delta deg(v_i) + \cdots \\ \cdots + & \sum_{v_j \notin star(v_i), v_j \in \mathcal{V}} \frac{deg(v_j) + \Delta deg(v_j)}{k_{ij} + \Delta k_{ij}} + \frac{deg(v_{N+1})}{k_{i,N+1}}) \cdots \\ \cdots + deg(v_{N+1}) + & \sum_{v_m \notin star(v_{N+1})} \frac{deg(v_m) + \Delta deg(v_m)}{k_{mj} + \Delta k_{mj}}, \end{aligned}$$

where $\Delta deg(v_i)$ is the change of degree at vertex v_i caused by the addition of a new vertex, and Δk_{mj} is the corresponding decrease in the shortest path between vertexes v_m and v_j.

It can be seen that adding a vertex always *increases* the complexity of the graph, as all perturbations are additive. It is therefore straightforward to capture the minimum or maximum perturbation that can be done by adding a vertex.

Theorem 5.4.1 *If $G \in \mathcal{G}^c_{N,\delta}$, then the complexity of the connected graph G is bounded above and below as*

$$C(\delta_N) \leq C(G) \leq C(\mathcal{K}_N), \tag{17}$$

where δ_N is the δ-chain on N vertices, and $\mathcal{K}_N$ is the complete graph.

Proof: We prove the theorem by induction. Suppose it is true that $C(G) \leq C(\mathcal{K}_N)$ for $G \in \mathcal{G}_{N,\delta}$. Note that for any vertex v_i in the graph, $deg(v_i) \leq N$. For $\mathcal{K}_N$, $deg(v_i) = N$ for all vertices. Therefore the maximum number by which any degree can be perturbed in $\mathcal{K}_N$ is 1. The perturbation will be maximized if all degrees are perturbed by 1. Similarly, in $\mathcal{K}_N$, $k_{ij} = 1$ for all pairs of vertices. The maximum perturbation will take place when the relation $k_{ij} = 1$ still holds for all pairs after addition of new vertex, i.e. all vertices are directly connected. It can be easily seen that this can only be accomplished by adding

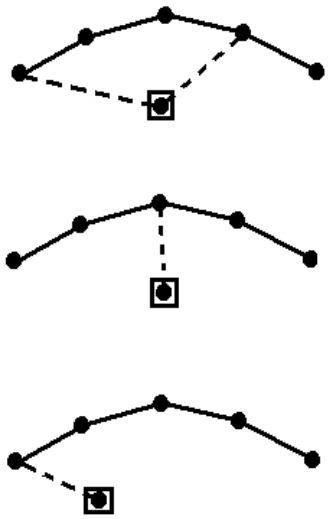


Figure 11: Different ways to add a new vertex to δ_5

edges between all vertices in K_N and the new vertex to make the graph $\mathcal{K}_{N+1}$. This proves that $C(G) \leq C(\mathcal{K}_N)$ for all N.

We repeat the induction argument for the lower bound as well. Suppose it is true that $C(\delta_N) \leq C(G)$ and we look at the perturbation equation of δ_N for minimum increase. (See Figure 11.) Since all terms in the perturbation equation are non-decreasing, it would be least perturbed, if each individual term is minimally increased. In order to produce a connected graph, $deg(v_{N+1}) \geq 1$. (If connectedness was not required, we would have added another vertex with 0 degree). For minimum increase, set $deg(v_{N+1}) = 1$. This would also mean that $\Delta deg(v_i) = 0$ for all v_i in δ_N except one. This corresponds to addition of exactly one edge to the old graph, δ_N. However this edge can be added to any of the N vertices. Note that this edge addition may disturb the shortest paths k_{ij} between node pairs v_i, v_j. (The paths cannot be lengthened by edge addition). If that happens, terms of the form $deg(v)/k$ will get bigger. The only way to avoid this is to add the edges to either end of the chain. Therefore $\Delta k_{ij} = 0$ for all $1 \leq i, j \leq N$. This also maximizes $k_{i,N+1}$ for all $1 \leq i \leq N$ so that $deg(v_{N+1})/k_{i,N+1} = 1/k_{i,N+1}$ are minimized for all $i \leq N$. This shows that if the edge is added to a vertex which is not an end point, it results in an addition of degrees as well as a decrease in k_{ij} for some vertices, again resulting in increase of complexity. Therefore, the

optimal way to add the edge is to add the edge at its ends, which results in another delta chain δ_{N+1}. ■

The consequence of this theorem is that the δ-chain can be thought of as the simplest formation that can be formed over a fixed number of agents. This perhaps explains why humans like to make queues and birds fly in V-formations, both of which are essentially δ-chains and require minimum coordination among individuals. We will use this result in the future to justify various δ-chaining algorithms that are part of our current investigations of connectivity graphs.

5.5 Open Problem II: Symmetry and Complexity

As described in the early part of this chapter, there is a distinction between *intrinsic* and *extrinsic* complexities [14] of structures. When studying a certain large-scale system according to the number of interactions in it, symmetry does not play a role since symmetric structures do not necessarily imply fewer interactions. This was the approach taken in this thesis, where the complexity measure is essentially a measure of intrinsic complexity. But, when studying problems of synthesis or fabrication, in which a process is iterated to build a complex structure, symmetry clearly implies lesser complexity [88]. The same approach should be taken when studying robot formations. The complexity of a robot formation that is already in place should not be affected by the potential symmetries of its corresponding connectivity graph. However, when studying the problem of *synthesizing* formations, it can be argued that it is easier to produce symmetric structures [58]. This is the case because in a completely symmetric formation, the same program can be executed by the different robots, which does not happen in asymmetric formations. Coming up with a complexity measure that deals with extrinsic complexity is therefore an open problem.

CHAPTER VI

GEOMETRIC STRUCTURE OF CONNECTIVITY GRAPHS

In this chapter we produce some results about the geometric structure of connectivity graphs. We will see that the graphs are made up of atomic graphs which, when combined in a certain way, produce more complex graphs. This decomposition will prove to be helpful in order to construct a *decentralized* method of obtaining a simplicial structure, buried inside the graph. This simplicial structure will consequently be used to associate a topological characterization to the connectivity graph.

6.1 What Does the Graph "Look Like"?

We start with the concept of an image of a graph. The image is what a graph would "look like" if drawn in the plane. Note that this is different from the concept of planar graphs [47] or imbedding graphs in $\mathbb{R}^2$ where edges are not necessarily mapped to straight lines.

Definition 6.1.1 (Image of a formation in $\mathbb{R}^2$) *If a given formation $\mathcal{F} = (x_1, x_2, \dots x_N) \in \mathcal{C}^N(\mathbb{R}^2)$ has the connectivity graph $G = (\mathcal{V}, \mathcal{E})\Phi_N(\mathcal{F}(t))$, then each edge $e_k = \{v_{k_1}, v_{k_2}\} \in \mathcal{E}$ can be mapped to $\mathbb{R}^2$ by a map $f_k : \mathbb{R} \to \mathbb{R}^2$ given by $f_k(s) = sx_{k_1} + (1-s)x_{k_2}$ for $s \in [0,1]$. We call the image of the mapping f_k, the* image *of the edge in $\mathbb{R}^2$. The* image of a formation*, $I_{\mathcal{F}} \subset \mathbb{R}^2$ can be defined as the union of the images of all edges in the connectivity graph of the formation.*

$$I_{\mathcal{F}} = \bigcup_{e_k \in \mathcal{E}} f_k([0,1]) \subset \mathbb{R}^2. \tag{18}$$

Note that the image is constructed by mapping each vertex v_i of the graph to its position x_i and each edge $e_k = \{v_{k_1}, v_{k_2}\}$ to a line segment $sx_{k_1} + (1-s)x_{k_2}$, for $s \in [0,1]$, in $\mathbb{R}^2$. If

it is clear from the context, what formation is under consideration, we will often write $I_{\mathcal{F}}$ as I_G, where $G = \Phi(\mathcal{F})$.

Sometimes it will be convenient to describe the image of a subgraph $H = (\mathcal{E}_H, \mathcal{V}_H)$ of the connectivity graph G of formation $\mathcal{F}$, where $\mathcal{E}_H \subset \mathcal{E}$ and $\mathcal{V}_H \subset \mathcal{V}$. In this case, we refer to the image of the subgraph H as

$$I_H = \bigcup_{e_k \in \mathcal{E}_H} f_k([0,1]) \subset \mathbb{R}^2,\ H \subseteq G = \Phi_N(\mathcal{F}) \in \mathcal{G}_{N,\delta}. \tag{19}$$

6.2 Crossing Generators and Δ-Amalgamations

Two edges $e_i, e_j \in \mathcal{E}$ of a graph are said to be *crossing* if $f_i(s) = f_j(t)$ for some $s, t \in (0,1)$ and the set $f_i([0,1]) \cap f_j([0,1])$ has dimension 0. According to this definition, edge intersection at some vertex of the two edges does not count as a crossing. The condition that the intersection set is of dimension 0, rules out edge intersections of collinear points. For convenience denote by $e_i \ltimes e_j$true, if $e_i, e_k \in \mathcal{E}$ are crossing edges. Given a formation $\mathcal{F}$ and its connectivity graph $\Phi(\mathcal{F}) = G = (\mathcal{E}, \mathcal{V})$, let $\mathcal{E}_\times \subseteq \mathcal{E}$ be the set of all crossing edges, and $\mathcal{V} \supseteq \mathcal{V}_\times = \{v \in \mathcal{V} \mid v \in e \text{ for some } e \in \mathcal{E}_\times\}$, then $H_\times = (\mathcal{E}_\times, \mathcal{V}_\times)$ is the subgraph of G made up of all crossing edges of the connectivity graph. We denote by $H_\triangle = (\mathcal{E} \setminus \mathcal{E}_\times, \mathcal{V})$, the subgraph of G consisting of all non-crossing edges so that $G = H_\times \cup H_\triangle$.

It should moreover be noted that the points in the image I_G can be categorized as smooth or non-smooth, where smoothness is defined in the setting of smooth manifolds. Any point x in the image that is not one of the robot positions $\{x_i\}_{i=1}^N$ or the crossing points is smooth, i.e. there always exists a small neighborhood $\mathcal{B}_\epsilon(x)$, and a homeomorphism $f : \mathcal{B}_\epsilon(x) \cap I_G \to \mathbb{R}$.

Proposition 6.2.1 *An image of a formation of 4 vertices has a pair of crossing edges only if its connectivity graph is isomorphic to either $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ as shown in Figure 12.*

Proof: From the Figures in 6, we see that only those connectivity graphs that are isomorphic to the graphs in Figure 12, namely those in 6.a 6.c and 6.d can be realized

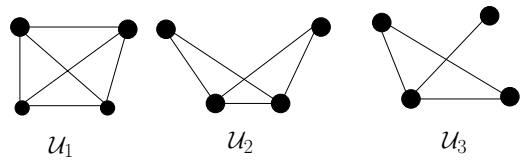


Figure 12: Crossing Generators.

with crossing edges. For all other graphs in Figure 6, any attempt to create an image with crossing edges results in a violation of the constraints that define these graphs. ∎

The three graphs $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ are called the *crossing generators* of all connectivity graphs. If G is a connectivity graph of a formation $\mathcal{F}$, and has two edges e_1, e_2 such that $e_1 \ltimes e_2 =$ true, then there always exists a subgraph of G that contains e_1 and e_2 and is isomorphic to one of the crossing generators $\mathcal{U}_1, \mathcal{U}_2$ or $\mathcal{U}_3$. We now define the following operation.

Definition 6.2.1 (Δ-Amalgamation of crossing generators) *If $\mathcal{U}_i, \mathcal{U}_j \in \mathcal{G}_4$ are two crossing generators, $H \subset \mathcal{U}_i$ and $H' \subset \mathcal{U}_j$ are subgraphs s.t. $H, H' \simeq \mathcal{K}_3$ (the complete graph on 3 vertices), and there is an isomorphism $\Delta : H \rightarrow H'$ between the respective subgraphs, then their amalgamation (as standard amalgamation of two graphs [47]) according to the isomorphism Δ is called a Δ-amalgamation, denoted by $\mathcal{U}_i *_\Delta \mathcal{U}_j$.*

In the context of connectivity graphs of formations, Δ - amalgamation [1] is used as a description for unions of the type, $\bigcup_i I_{G_i}$, where each $G_i \subseteq G$, each G_i is a valid connectivity graph in $\mathcal{G}_{4,\delta}$ and each $G_i \simeq \mathcal{U}_j$ for some $1 \leq j \leq 3$. As explained in later sections, such amalgamations help characterize the geometrical constraints that give rise to crossing edges and non-smooth points in the image of a connectivity graph. They are used as an encoding method to describe the subgraph $H_\times = (\mathcal{E}_\times, \mathcal{V}_\times) \subseteq G$ made up of crossing edges. Therefore, we explain next, the kind of Δ-amalgamations appropriate to facilitate this encoding process, and hence the concept of a *well-defined* Δ-amalgamation.

It should be emphasized that there can be several ways to obtain Δ- amalgamations between any two crossing generators, depending on the choice of H and H'. Examples of such amalgamations are illustrated in the Figure 13, where the two crossing generators,

[1] The letter Δ in the notation is used to emphasize that all amalgamations are done over triangular subgraphs.

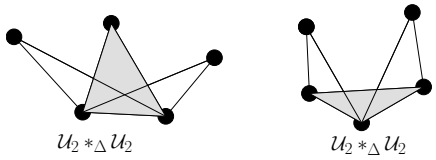


Figure 13: Different Δ-amalgamations of $\mathcal{U}_2$ and $\mathcal{U}_2$. The subgraphs involved in the Δ-amalgamations are shaded in gray.

both isomorphic to $\mathcal{U}_2$ produce two different Δ- amalgamations. The subgraphs involved in the Δ-amalgamations are shaded for ease of visualization. It should also be emphasized that every such Δ- amalgamation is not desirable. A sufficient condition for a Δ-amalgamation $G_{i_1} *_\Delta G_{i_2}$ to be *well- defined* is that $G_{i_1} *_\Delta G_{i_2}$ be a valid connectivity graph in $\mathcal{G}_{5,\delta}$. Generalizing this for an arbitrary number of amalgamations, if

$$\underbrace{G_{i_1} *_\Delta G_{i_2} *_\Delta \ldots G_{i_k}}_{(k-1)-amalgamations} \in \mathcal{G}_{4+k,\delta} \tag{20}$$

then the operation is well defined. Now consider the examples in the left-half of Figure 14. The three examples can be described by $\mathcal{U}_1 *_\Delta \mathcal{U}_1$, $\mathcal{U}_2 *_\Delta \mathcal{U}_2$ and $\mathcal{U}_3 *_\Delta \mathcal{U}_2 *_\Delta \mathcal{U}_3$ from top to bottom, and can nicely encode their respective subgraphs, that contain the crossing edges. The graphs in Figures 14.a and 14.c are not valid connectivity graphs. The constraints dictated by the respective graphs cannot result in valid formations in the configuration space. Similarly, the Δ-amalgamation drawn in Figure 14.e may or may not be valid. However, if we consider the graphs drawn on the right in Figures 14.b, 14.d and 14.f, the graphs are not only valid connectivity graphs but their crossing edges can also be encoded by the Δ-amalgamations described earlier. Based on this observation we define the following.

Definition 6.2.2 (Closure of a Δ-amalgamation) *Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ denote two subgraphs of G. Let $\bar{V} = V_1 \cup V_2$, $\bar{E} = \{(v_i, v_j) \mid v_i, v_j \in \bar{V}\}$ and $\bar{G} = (\bar{V}, \bar{E})$. If $G_1 *_\Delta G_2$ is a Δ-amalgamation of G_1 and G_2, then the* Closure *of $G_1 *_\Delta G_2$ is defined as*

$$\overline{G_1 *_\Delta G_2} = (G_1 *_\Delta G_2) \cup \bar{G}.$$

In Figure 14, the graphs drawn on the right are closures of the graphs drawn on the left. If $\bar{G}\emptyset$ then the Δ-amalgamation is equal to its closure. The graphs shown in Figure 13 and

Figure 14.e can be such examples. We now give the following definition of a *well-defined* Δ-amalgamation.

Definition 6.2.3 (Well-defined Δ-amalgamation) *A Δ-amalgamation of two subgraphs G_1, G_2 of a graph $G = H_\times \cup H_\Delta$, denoted by $G_1 *_\Delta G_2$ is well defined if*

$$\overline{G_1 *_\Delta G_2} = (G_1 *_\Delta G_2) \cup \bar{G} \in \mathcal{G}_{5,\delta},$$

and $\bar{E} \cap E_\times = \emptyset$.

In other words, a Δ-amalgamation is well-defined if it does not differ from its closure by any crossing edges. The Δ-amalgamation operation can be repeated to generate a whole family of graphs from the crossing generators, and is well-defined if the resulting graph's closure does not differ by any crossing edges. If we let $\Sigma = \sigma_1.\sigma_2.\ldots.\sigma_K$ be a finite string defined over $\{1, 2, 3\}$, then we denote a member of this family as:

$$\mathcal{G}_\Sigma = \mathcal{U}_{\sigma_1} *_\Delta \mathcal{U}_{\sigma_2} *_\Delta \ldots *_\Delta \mathcal{U}_{\sigma_K} \tag{21}$$

If we have repeated Δ-amalgamations of subgraphs of a connectivity graph, as in (20), there always exists a finite string Σ such that

$$\mathcal{G}_\Sigma \simeq G_{i_1} *_\Delta G_{i_2} *_\Delta \ldots G_{i_k} \subseteq \overline{G_{i_1} *_\Delta G_{i_2} *_\Delta \ldots G_{i_k}} \in \mathcal{G}_{4+k,\delta} \tag{22}$$

Each well-defined repeated Δ-amalgamation, as defined in (22), is called an *Atomic Crossing Graph*. Let $I_{\mathcal{G}_\Sigma}$ denote the image of a atomic crossing graph by referring to its isomorphic graph $\mathcal{G}_\Sigma$, when the details of the Δ-amalgamations is clear from context.

Proposition 6.2.2 *There exist a set of atomic crossing graphs $\{\mathcal{G}_{\Sigma_j}\}_{j\in J}$ such that*

$$I_{H_\times} \subset \bigcup_{j\in J} I_{\mathcal{G}_{\Sigma_j}} \subseteq I_G \subset \mathbb{R}^2, \tag{23}$$

where J is some finite indexing set, and each $x \in \left(I_{\mathcal{G}_{\Sigma_i}} \cap I_{\mathcal{G}_{\Sigma_j}}\right) \setminus \{x_k\}_{k=1}^N$ for $i, j \in J$, is smooth.

Proof: With slight abuse of notation, let $e \in G = (\mathcal{E}, \mathcal{V})$ denote that $e \in \mathcal{E}$. If $e_i \ltimes e_j =$ true, then denote by $H_q(e_i, e_j)$ the subgraph of G such that $H_q(e_i, e_j) \simeq \mathcal{U}_q$ for some

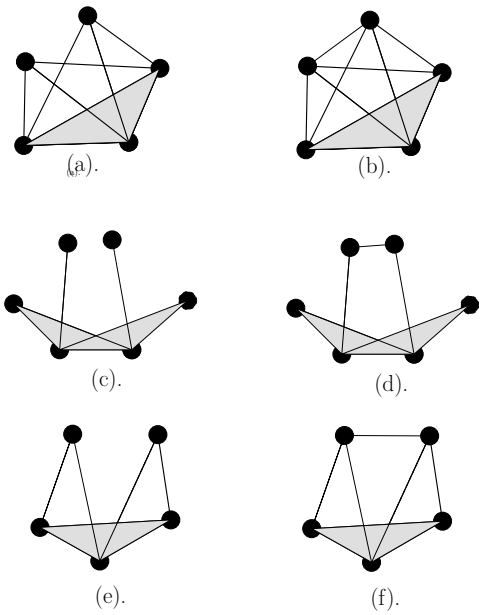


Figure 14: Examples of Δ-amalgamations (a,c,e) and their closures (b,d,f).

$1 \leq q \leq 3$ and $e_i, e_j \in H_q(e_i, e_j)$. Also denote by $\mathcal{E}_{\Sigma_j}$ the set of edges for the graph $\mathcal{G}_{\Sigma_j}$. We now give the following algorithm to provide a constructive way of obtaining the atomic crossing graphs.

Algorithm 6.2.1 .

A $j \leftarrow 0$

B $\mathcal{E}_\times = \{e \in \mathcal{E} \mid e \ltimes e' = \textit{true for some } e' \in \mathcal{E}\}$

C *while* $\mathcal{E}_\times \neq \emptyset$

1. $j \leftarrow j + 1$
2. $k \leftarrow 1$
3. *Pick* $e_m \in \mathcal{E}_\times$
4. *Pick* $e_p \in \mathcal{E}_\times$ *such that* $e_m \ltimes e_p = \textit{true}$
5. $\mathcal{E}_\times \leftarrow \mathcal{E}_\times \setminus \{e_p, e_m\}$
6. $\sigma_k = \arg\min_{1 \leq q \leq 3} (H_q(e_m, e_p))$

7. $\Sigma_j \leftarrow \sigma_k$

8. $\mathcal{G}_{\Sigma_j} \leftarrow H_{\sigma_k}(e_m, e_p)$

9. $\mathcal{E}_{j_\circ} = \{e \in \mathcal{E}_\times \mid e \in H_{\sigma_k}(e_m, e_p)\}$

10. $\mathcal{E}_\times \leftarrow \mathcal{E}_\times \setminus \mathcal{E}_{j_\circ}$

11. $\mathcal{E}_{j_\times} = \{e \in \mathcal{E}_\times \mid e \ltimes e_l = \textit{true for some } e_l \in \mathcal{E}_{\Sigma_j}\}$

12. *while* $\mathcal{E}_{j_\times} \neq \emptyset$

    (a) *Pick* $e_r \in \mathcal{E}_{j_\times}$

    (b) *Pick* $e_s \in \mathcal{E}_{\Sigma_j}$ *such that* $e_r \ltimes e_s = \textit{true}$

    (c) $\mathcal{E}_\times \leftarrow \mathcal{E}_\times \setminus \{e_r, e_s\}$

    (d) $\sigma_{k+1} = \arg\min_{1 \leq q \leq 3} (H_q(e_r, e_s))$

    (e) $\Sigma_j \leftarrow \Sigma_j \sigma_{k+1} = \sigma_1 \sigma_2 \ldots \sigma_k \sigma_{k+1}$

    (f) $\mathcal{G}_{\Sigma_j} \leftarrow \mathcal{G}_{\Sigma_j} *_\Delta H_{\sigma_{k+1}}(e_r, e_s)$

    (g) $\mathcal{E}_{j_\circ} = \{e \in \mathcal{E}_\times \mid e \in \mathcal{E}_{j_\times} \wedge e \in H_{\sigma_{k+1}}(e_r, e_s)\}$

    (h) $\mathcal{E}_\times \leftarrow \mathcal{E}_\times \setminus \mathcal{E}_{j_\circ}$

    (i) $\mathcal{E}_{j_\times} = \{e \in \mathcal{E}_\times \mid e \ltimes e_l = \textit{true for some } e_l \in \mathcal{E}_{\Sigma_j}\}$

    (j) $k \leftarrow k + 1$

13. *end*

D end

The algorithm can be explained as follows. First get the set $\mathcal{E}_\times$ of all crossing edges of the graph (Step B). Execute the algorithm until all crossing edges are taken care of, i.e. removed from $\mathcal{E}_\times$. Start by picking randomly a pair of crossing edges e_m and e_p in $\mathcal{E}_\times$. Next, obtain the maximal subgraph $H_{\sigma_k}(e_m, e_p)$ of G that contains the two edges, such that it is isomorphic to one of the crossing generators (Steps C.6 $\cdots$ C.8). This gives the starting point for generating an atomic crossing graph. Delete all edges in $\mathcal{E}_\times$ that are also contained in $H_{\sigma_k}(e_m, e_p)$. Next, find all edges in $\mathcal{E}_\times$ that cross with any edge in $H_{\sigma_k}(e_m, e_p)$ (Step C.11). Now, pick an edge from this set and one of its crossing edges, say e_r and e_s.

The pair of crossing edges will again span a maximal subgraph $H_{\sigma_{k+1}}(e_r, e_s)$ isomorphic to one of the crossing generators (Step C.12.d). This new subgraph will always intersect the previously found subgraph $H_{\sigma_k}(e_m, e_p)$ over a subgraph isomorphic to $\mathcal{K}_3$. This lets us glue the two subgraphs together by a Δ-amalgamation operation (Step C.12.f). Again, remove the crossing edges that are part of the newly found edge set $\mathcal{E}_{j\circ}$ (Steps C.12.g, k). Next, find the crossing edges that cross any edge of the resulting Δ-amalgamation. Now, repeat once again the process of picking randomly a pair of crossing edges, finding their crossing generator and gluing it by repeated Δ-amalgamations. This process continues until no crossing edge remains that crosses any edge of the resulting repeated Δ-amalgamation. Delete all crossing edges that are used in this construction. In this way we get our first atomic crossing graph $\mathcal{G}_{\Sigma_1}$. Now, go back to the main loop and repeat the same process for any remaining crossing edges to get a whole family of atomic crossing graphs $\mathcal{G}_{\Sigma_1}, \mathcal{G}_{\Sigma_2} \cdots \mathcal{G}_{\Sigma_j}$, until $\mathcal{E}_\times \emptyset$.

Since there are only finitely many crossing edges, this algorithm always terminates in less than $|\mathcal{E}_\times|/2$ number of Δ-amalgamations. By executing this algorithm, it is clear that all crossing edges are eventually absorbed into one of $\mathcal{G}_{\Sigma_i}$ in a finite number of steps so that $H_\times \subset \cup_{j\in J}\mathcal{G}_{\Sigma_j}$, and hence $I_{H_\times} \subset \bigcup_{j\in J} I_{\mathcal{G}_{\Sigma_j}} \subseteq I_G \subset \mathbb{R}^2$. ∎

The above discussion gives a decomposition of connectivity graphs in terms of crossing and non-crossing edges. An example of such decomposition is given in Figure 15. These properties will become useful for obtaining a simplicial representation of connectivity graphs, which will subsequently help in understanding the "topological shape" of formations as discussed in the following section.

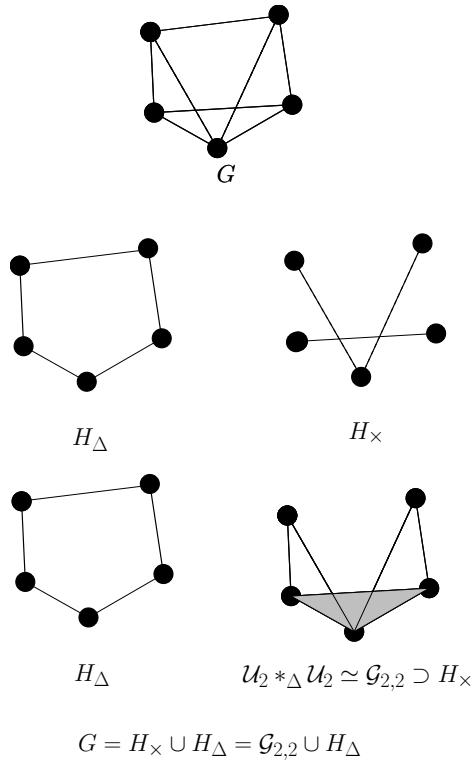


Figure 15: A connectivity graph and its decomposition in terms of atomic crossing graphs.

CHAPTER VII

TOPOLOGY OF FORMATIONS I: RIPS-VIETORIS COMPLEXES AND TOPOLOGICAL INVARIANTS

We first describe informally, as to what simplicial complexes are and how to generalize a connectivity graph to a simplicial complex. A simplicial complex is a finite collection of objects called simplices. Given a collection of points, a simplex is an unordered non-repeating subset of this collection. The number of points in a simplex minus one, defines its dimension. A simplex of dimension k is usually called a k-simplex. The faces of a k-simplex consist of all $(k-1)$-simplices that can obtained from the points in the k-simplex alone. The simplices are usually given an orientation, as depicted in Figure 17. A simplicial complex can be made up of simplices of several dimensions. However, the simplices of various dimensions have to satisfy this property: Whenever a simplex lies in the collection then so does each of its faces, and whenever two simplices intersect, they do so in a common face. See Figure 16 for some counter examples. Triangulated surfaces form a concrete example,

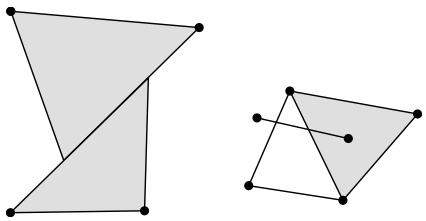


Figure 16: Non-examples of simplicial complexes. The discs are the 0-simplices, lines connecting 0-simplices represent the 1-simplices, and the shaded triangles depict the 2-simplices. [Left] The shaded triangles do not have valid 1-simplices as their faces. Furthermore, they do not intersect in a 1-simplex. [Right] One of the 1-simplices does not have a common face with the 2-simplex and two other 1-simplices.

where the vertices of the triangulation correspond to 0-simplices, edges correspond to 1-simplices, and faces correspond to 2-simplices. Graphs are also valid simplicial complexes that are composed of 0-simplices (nodes) and 1-simplices (edges) only. The ordering of the

vertices corresponds to an orientation. An important point to remember is that any abstract simplicial complex on a (finite) set of points has a geometric realization in some $\mathbb{R}^d$, for large enough d. Therefore, it is important to distinguish between the abstract combinatorial object, the simplicial complex and its realization as a set of points in a certain Euclidean space. After this informal introduction, we provide a rigorous introduction to simplicial complexes and simplicial homology. Since, this subject is not found in the reportage of electrical engineering literature, a more detailed introduction to various concepts of algebraic topology has been provided in the Appendix. See [5, 82, 55] for a thorough yet elementary introduction to this subject. For more advanced topics, see [21, 49].

7.1 Simplicial complexes and Simplicial Homology

Given a set of points V, a k-simplex is an unordered subset $\{v_0, v_1, \ldots, v_k\}$ where $v_i \in V$ and $v_i \neq v_j$ for all $i \neq j$, see Figure 17. The *faces* of this k-simplex consist of all $(k-1)$-

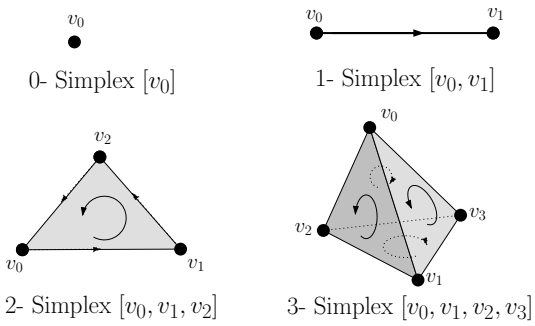


Figure 17: Oriented simplices of dimension zero through three.

simplices of the form $\{v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k\}$ for $0 \leq i \leq k$. A *simplicial complex* is a collection of simplices which is closed with respect inclusion of faces.

Homology is an algebraic procedure for counting 'holes' of various types. There are numerous variants of homology. The topological objects most relevant to this work are the the simplicial complexes. For these simplicial complexes, we describe the variant called as *simplicial homology*. For a more thorough introduction to simplicial and singular homology theories, see the Appendix.

Let X denote a simplicial complex. The homology of X, denoted $H_*(X)$, is a sequence of vector spaces $\{H_k(X) : k = 0, 1, 2, 3 \ldots\}$, where $H_k(X)$ is called the *k-dimensional homology*

of X. The dimension of $H_k(X)$, called the k^{th} *Betti number of* X, counts the number of different holes in the space X that can be sensed by using sub-complexes of dimension k. Readers familiar with the Euler characteristic of a triangulated surface will not find it odd that a certain way of counting the simplices yields a topological invariant. For simplicial complexes X, the Euler characteristic $\chi(X)$ is the alternating sum of the Betti numbers.

$$\chi(X) = \sum_i (-1)^i \text{ rank } H_i(X).$$

Let X denote a simplicial complex. Define for each $k \geq 0$, the vector space $C_k(X)$ to be the vector space whose basis is the set of *oriented* k-simplices of X; that is, a k-simplex $\{v_0, \ldots, v_k\}$ together with an order type denoted $[v_0, \ldots, v_k]$ where a change in orientation corresponds to a change in the sign of the coefficient. More precisely,

$$[v_0, \ldots, v_j, \ldots, v_i, \ldots, v_k] = -[v_0, \ldots, v_i, \ldots, v_j, \ldots, v_k].$$

For k larger than the dimension of the simplicial complex X, $C_k(X) = 0$. The *boundary map* is defined to be the linear transformations $\partial_k : C_k(X) \to C_{k-1}(X)$ which acts on basis elements $[v_0, \ldots, v_k]$ via

$$\partial_k [v_0, \ldots, v_k] := \sum_{i=0}^{k} (-1)^i [v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k], \tag{24}$$

as illustrated in Figure 18.

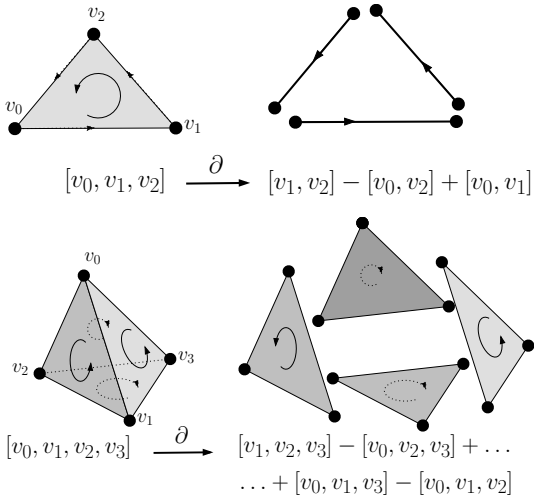


Figure 18: The boundary operator on a 2-simplex [top] and a 3-simplex [bottom].

The boundary maps and the vector spaces give rise to a *chain complex*

$$\cdots \xrightarrow{\partial_{k+2}} C_{k+1}(X) \xrightarrow{\partial_{k+1}} C_k(X) \xrightarrow{\partial_k} C_{k-1}(X) \cdots \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \xrightarrow{\partial_0} 0$$

Now, consider the following two subspaces of $C_k(X)$: the *cycles* (those sub-complexes without boundary) and the *boundaries* (those sub-complexes which are themselves boundaries). More precisely, the k-cycles are defined by

$$Z_k(X) = \ker\,(\partial_k : C_k(X) \to C_{k-1}(X)), \tag{25}$$

and the k-boundaries by

$$B_k(X) = \mathrm{im}(\partial_{k+1} : C_{k+1}(X) \to C_k(X)). \tag{26}$$

A simple calculation (see Appendix) demonstrates that $\partial_{k+1} \circ \partial_k = 0$. In other words, the boundary of a chain has an empty boundary. From this , it follows that $B_k(X) \subset Z_k(X)$. The k-cycles in X are the basic objects which count the presence of a 'hole of dimension k' in X. Two cycles α_1 and α_2 in $Z_k(X)$ are *homologous* if $\alpha_1 - \alpha_2 \in B_k(X)$. In other words, two cycles are homologous if their difference is a boundary. The k-dimensional *homology* of X, denoted $H_k(X)$ is the quotient vector space,

$$H_k(X) = \frac{Z_k(X)}{B_k(X)}. \tag{27}$$

Specifically, an element of $H_k(X)$ is an equivalence class of homologous k-cycles. The equivalence class of homologous cycles of $\alpha \in Z_k(X)$ is denoted by $[\alpha]$. This quotient space inherits the structure of a vector space in a natural way: $[\beta_1] + [\beta_2] = [\beta_1 + \beta_2]$ and $c[\beta] = [c\beta]$ for $c \in \mathbb{Z}$. Homology $H_*(X)$ is a topological invariant of X. It is an invariant of *homotopy type*. For a definition of homotopy type and a proof of homotopy invariance, please see the Appendex.

Consider two simplicial complexes X and Y. Let $f : X \to Y$ be a continuous map that takes each k-simplex of X to an n-simplex of Y, where $n \leq k$. Then, the map f induces a linear transformation $f_\# : C_k(X) \to C_k(Y)$. As shown in the Appendex, $f_\#$ takes cycles to cycles and boundaries to boundaries. Therefore, there is a well-defined linear transformation

$$f_* : H_k(X) \to H_k(Y) \quad : \quad f_* : [\xi] \mapsto [f_\#(\xi)].$$

This is called the *induced homomorphism* of f on $H_*(X)$. A detailed discussion of the induced homomorphisms has been outlined in the Appendix.

7.2 Simplicial Complexes in Networks

So how do we go from connectivity graphs to a simplicial complexes? We construct what is known as a Rips-Vietoris complex. This type of a simplicial complex goes back to the early papers of Vietoris on the foundations of homology theory [102]. Similar objects were reinvented by Rips in the 1980's in the context of geometric group theory and have been used extensively since. The nodes and edges of the Rips complex are precisely those in the corresponding connectivity graph. We add simplices of higher dimension in the following way. Any time we find a triangle in the communication graph, we fill in an abstract 2-simplex. Any time we see a complete subgraph on $k+1$ vertices, we fill it in with an abstract k-simplex. Therefore, each k-simplex in the Rips complex corresponds to to unordered $(k+1)$-tuples of points in the nodes which are pairwise within the communication range of each other. We give a precise definition below.

Definition 7.2.1 *Given a set of points $\mathcal{X} = \{x_\alpha\} \subset \mathbb{R}^n$ in Euclidean n-space and a fixed radius ϵ, the* Vietoris-Rips complex *of $\mathcal{X}$, $\mathcal{R}(\mathcal{X})$, is the abstract simplicial complex whose k-simplices correspond to unordered $(k+1)$-tuples of points in $\mathcal{X}$ which are pairwise within Euclidean distance ϵ of each other.*

It is often helpful to visualize this complex as drawn in Figure 19. The bottom drawing depicts the agents in a plane as well as their connectivity graph. The communication radii are depicted as dotted circles for two agents for the sake of clarity. The top drawing shows the Rips complex induced from the connectivity graph. Some of the higher dimensional simplices are colored for distinction. The red simplex is a of dimension 4, the blue and green simplices are of dimension 3. Intuitively, the Rips complex lifts the connectivity graph up to a higher dimensional space that allows the study of simultaneous relationships between more than 2 agents. In a connectivity graph, the edges allowed the study of pairwise relations only. Therefore, the Rips complex is in some sense a more powerful way of capturing the spatial relations between agents when compared to a connectivity graph.

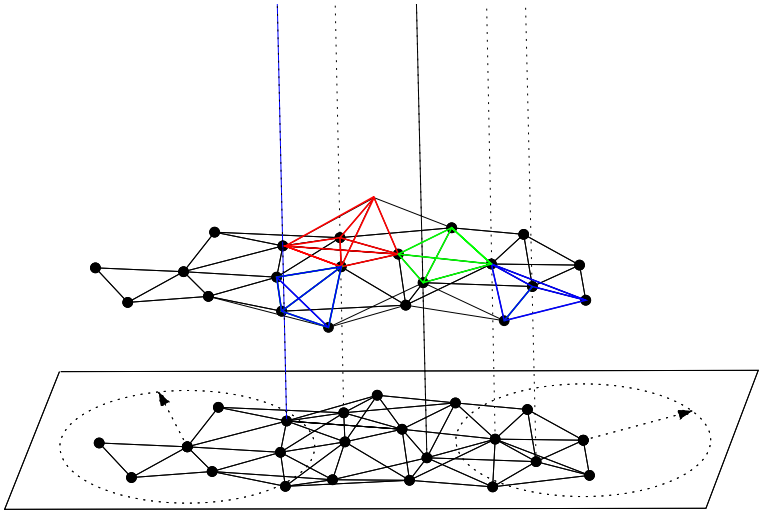
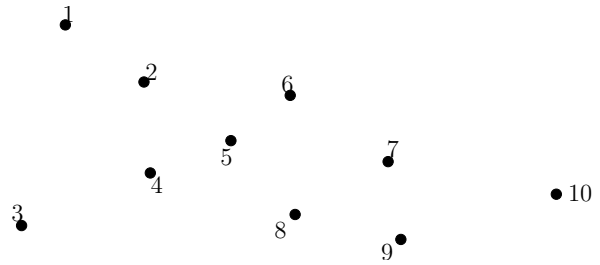


Figure 19: A visualization method for connectivity graphs and their Rips complexes.

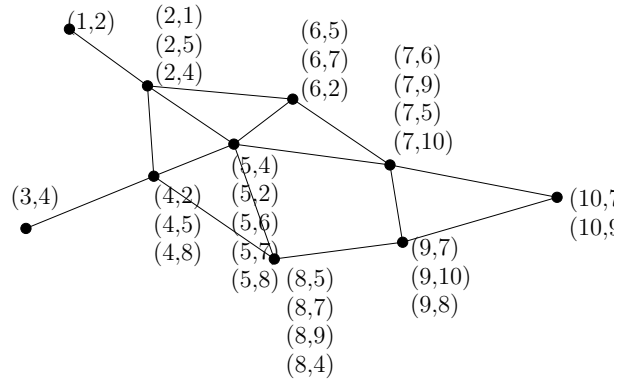
7.3 A Network Protocol for Constructing the Rips Complex

It is also worth mentioning that in the case of a real life multi-agent system equipped with real radios, the Rips complex can be constructed by using a very simple synchronous broadcast protocol. The objective is that each agent needs to be aware of what simplices it is a part of, and what other agents share those simplices. Suppose that each agent carries a unique identification tag. Also, it is capable of communicating its identification to its neighboring agents along with some other information of interest. Each agent also maintains an array of lists of identification tags, each list corresponding to a simplex that the agent is a part of. See Figure 20 for an example.

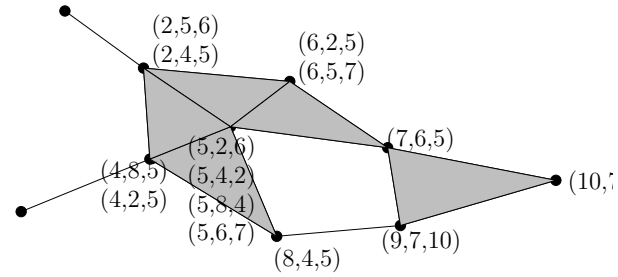
In the start, each agent is aware of its own identification alone. Therefore the first entry is the identification tag of the agent alone. This generates the 0-simplices. Now the agents simultaneously broadcast their identification. The agents within the communication radius receive this information and append this the received tags paired with their own tags in their respective arrays. This generates the 1-simplices or the edges in the simplicial complex. Once again, the agents broadcast, but this time the list of edges that possess. After reception, each agent compares the received list of edges to its own array of edges and searches for a cycle. If it finds one, it appends each three cycle to its array, thus generating the 2-simplices. In the next broadcast, the lists of 2-simplices is shared between neighboring



(a) Before Broadcast: 0-simplices are discovered.



(b) First Broadcast: 1-simplices are discovered.



(c) Second Broadcast: 2-simplices are discovered.

Figure 20: Discovery of the Rips complex via a broadcast protocol.

agent, thereby discovering 3-simplices. In this way, the all simplices of dimension k or lower are discovered in k broadcasts.

7.4 Homology Groups in Networks

We now interpret the homology groups computed from the Rips-complexes corresponding to different networks. For a Rips-complex $\mathcal{R}$, the dimension of the zeroth homology group $H_0(\mathcal{R})$ counts the number of connected components of the network. Clearly, $H_0(\mathcal{R}) \simeq \mathbb{Z}$ if and only if the corresponding connectivity graph is connected.

The homology group $H_1(\mathcal{R})$ counts the number of network holes in the connectivity

graph. A network hole is defined as a cycle of 4 edges or more, in which there exist at least 4 nodes, that are not spanned by any combination of 2-simplices. An example of a network hole in depicted in the network drawn in Figure 21. The Rips-complex has two network holes, one on the top, and one in the lower left corner. Therefore $H_1(\mathcal{R}) \simeq \mathbb{Z} \oplus \mathbb{Z}$ with two generators. The network hole at the top can be detected by one of the four homologous cycles drawn with thick lines. We will discuss the role of minimal generators and network hole detection in a later chapter. Finally, we study the higher homology groups in Rips

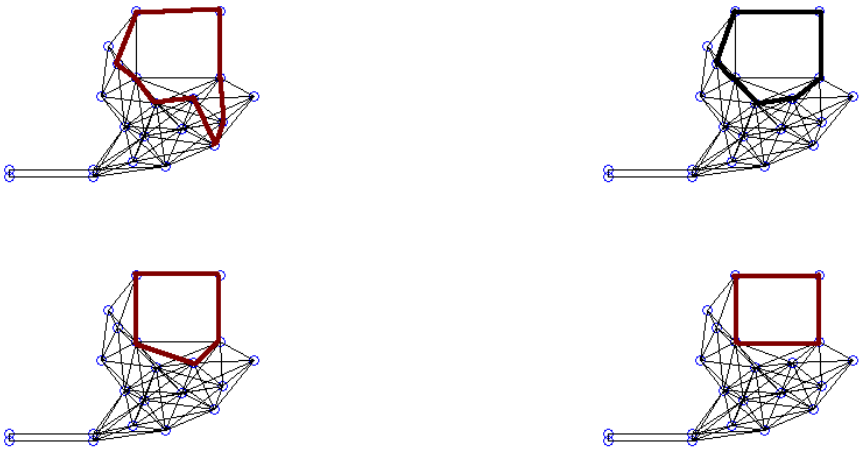


Figure 21: Four homologous cycles of $H_1(\mathcal{R})$, all detecting the same network hole.

complexes. An example of a connectivity graph, whose Rips-complex $\mathcal{R}_2$ has a non-trivial second homology, is drawn in Figure 22. The connectivity graph is drawn on the left. At first one might be mislead into thinking that that the first homology group, corresponding to its Rips-complex $\mathbb{R}_2$, is non-trivial and the thickened 1-cycle drawn in blue is a network hole. On more careful observation, one should observe that this cycle is in fact homologous to the zeros 1-cycle. The second homology group of this Rips-complex is however not trivial. The Rips-complex is a triangulated representation of a sphere S^2, which can be appreciated from the two drawings in the middle. Therefore,

$$H_k(\mathcal{R}_2) \simeq H_k(S^2) \simeq \begin{cases} \mathbb{Z}, \text{for } k = 0, 2; \\ 0, \text{ otherwise.} \end{cases}$$

The fact that $H_1(\mathcal{R}_2) \simeq 0$ can be understood more clearly from the right most drawing,

where the 1-cycle at the equator, is shown to be homologous to a point. This is true for any 1-cycle on the sphere. Similar constructions can be obtained where the other higher

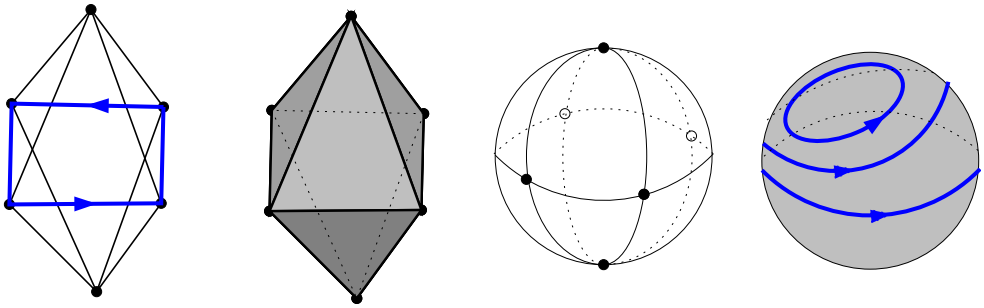


Figure 22: A connectivity graph [left] having the Rips-complex [center-left]with trivial first homology, but a non-trivial second homology group. It has the same homotopy type as that of a sphere [right].

homology groups are non-trivial. For the applications discussed in this work, the first and second homology groups have been found to be most useful.

CHAPTER VIII

TOPOLOGY OF FORMATIONS II: DECENTRALIZED METHODS FOR PRODUCING TRIANGULATIONS

Let us now look more closely into the image of a graph as explained in a previous chapter. Recall that we are interested in how a graph would "look like" if drawn in the plane. We are particulary interested in the question of whether the "image" of the graph looks like a triangulation, when drawn in the plane. If not, can we remove some edges from the graph to make it a triangulation? And if we can do that too, is there a way of doing it in a decentralized manner? The motivation for studying the question is that triangulations are perhaps the simplest objects, i.e. simplicial complexes, for which topological invariants can be computed. Therefore, a triangulation corresponding to a connectivity graph lets us study the "shape" of a configuration from a topological perspective. In particular, one can study if the presence of holes creates an obstacle towards achieving a global objective. Holes in networks have a relevance to the study of coverage and routing problems in sensor networks. Is there a relevance to coordination problems as well? We answer this question in the affirmative by giving applications in later chapters.

Figure 23 captures the intuition behind the utility of working on this problem using simplicial complexes. In the first step, the connectivity graph is lifted "up" into its Rips complex. We then find a surface in this lifted complex, which when projected down on the plane, gives a valid triangulation corresponding to the connectivity graph. This is now a valid simplicial complex, even when realized in a plane and ready for the computation of topological invariants using very simple methods. It is easy to see that the main hinderance in getting a planar triangulation from a graph is the presence of edges that intersect. A triangulation is a simplicial complex realized in $\mathbb{R}^2$. Therefore, the crossing edges violate the condition described above. In this Chapter we give a detailed method of removing the crossing edges in a decentralized manner.

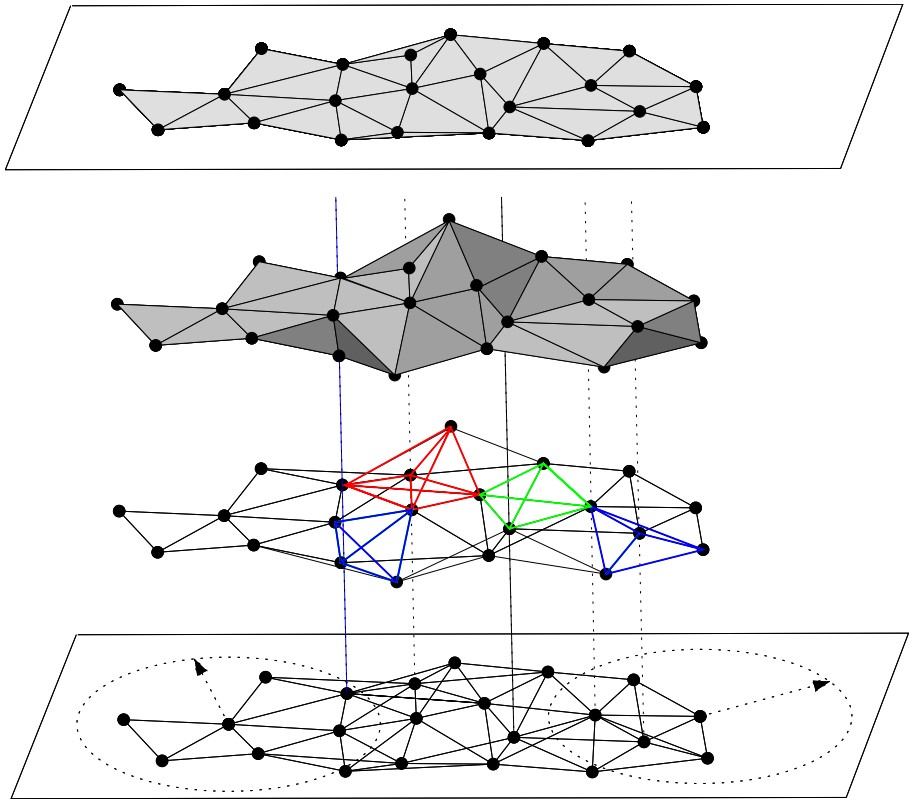


Figure 23: Obtaining triangulations from connectivity graphs. [Bottom] Agents in a plane and their connectivity graph. [Center Bottom] The Rips complex induced from the connectivity graph. [Center Top] A surface in the Rips complex corresponding to the triangulation of the connectivity graph. [Top] The triangulation of the connectivity graph.

8.1 Simplicial Complexes as Realizations in $\mathbb{R}^k$

It is a well known fact from algebraic topology [49] that the study of topological shapes of compact closed manifolds is synonymous to the study of triangulations of manifolds. As discussed in the previous chapter, these triangulations are examples of simplicial complexes [5]. For the sake of clarity, we here give some background to simplicial complexes as realization in a Euclidean space. Note that one should distinguish between the simplicial complex as an abstract combinatorial object and its realization in some Euclidean space $\mathbb{R}^k$.

As a realization in $\mathbb{R}^k$, a *k-simplex* is the smallest convex set that contains $k+1$ points in general position A set of points are in general position if any subset of them spans a strictly smaller hyperplane in $\mathbb{R}^k$. A finite collection of simplices (of dimension less than or equal to n) in $\mathbb{R}^n$ is a *simplicial complex* if whenever a simplex lies in the collection then

so does each of its faces, and whenever two simplices intersect, they do so in a common face. The dimension of a simplicial complex is equal to the largest of the dimensions of its simplices. In line with the terminology developed in the preceding chapters, the simplicial complex defined in this manner can be thought of as a *realization* of the corresponding abstract combinatorial object. From this point onwards, we only consider realizations in $\mathbb{R}^2$.

A simplicial complex of dimension 1, as a realization in $\mathbb{R}^2$, can be thought of as a graph whose image has no crossing edges. The only non-smooth points in the complex are the images of the vertices. Therefore, if some appropriate crossing edges are removed from a connectivity graph, its image is a well-defined simplicial complex, as realized in $\mathbb{R}^2$. Proposition 6.2.2 leads to some conclusions along these lines. All points in the image $I_{H_\triangle}$ are smooth except the vertex points. This makes $I_{H_\triangle}$ a well defined simplicial complex of dimension 1. Therefore the problem of obtaining a simplicial representation for the graph is reduced to finding one for $I_{H_\times}$. If the image of each atomic crossing graph can be converted into a simplicial complex, by removal of images of crossing edges, then the union of the individual simplicial complexes would be a well-defined simplicial complex, as guaranteed by Proposition 6.2.2. Therefore the problem of obtaining a simplicial complex from a connectivity graph can be solved if each sub-problem of obtaining a simplicial complex for each atomic crossing graph can be solved.

Definition 8.1.1 (Simplicial Subgraph) *Every connectivity graph $G \in \mathcal{G}_{N,\delta}$ has at least one subgraph G_s which induces a well defined simplicial complex K_{G_s}. We call G_s a* simplicial subgraph *of G.*

Every edge induces a 1-simplex in K_{G_s}. The subgraph of non-crossing edges $H_\triangle \subseteq G$, is also an example of a simplicial subgraph of G.

Definition 8.1.2 (Maximal Simplicial Subgraph) *A subgraph $G^* \subset G$ is said to be a maximal simplicial subgraph of G if there does not exist a simplicial subgraph of G that properly contains G^*.*

Before developing a method to obtain this maximal simplicial subgraph, a few points should be noted:

1. In order to preserve maximality, the removal of any non-crossing edge in the graph is not allowed.

2. Care has to be taken during the removal of crossing edges, as the removal of one crossing edge may result in the removal of all non-smooth points on another crossing edge, making the later non-crossing. Therefore the order in which crossing edges are removed is important.

3. The maximal simplicial subgraph of a connectivity graph is not unique and depends on the order in which crossing edges are removed.

8.2 Algorithms for Creating Realizations in $\mathbb{R}^2$

We begin by considering the problem of obtaining maximal simplicial subgraphs of atomic crossing graphs.

Proposition 8.2.1 *There exists an algorithm to obtain a maximal simplicial subgraph of every atomic crossing graph.*

Proof: Let an atomic crossing graph $G = (\mathcal{E}, \mathcal{V})$ be isomorphic to G_Σ, as given by 22 or obtained by executing Algorithm 6.2.1, so that the string $\Sigma = \sigma_1\sigma_2\cdots\sigma_K$ gives the order of Δ-amalgamations in the atomic graph,

$$G_\Sigma \simeq G = G_1 *_\Delta G_2 *_\Delta \ldots G_K \in \mathcal{G}^c_{4+K,\delta},$$

where $G_k \simeq \mathcal{U}_{\sigma_k}$ for $1 \leq k \leq K$. Now execute the following algorithm.

Algorithm 8.2.1 .

1. $\mathcal{E}^* \leftarrow \mathcal{E}$

2. $G^* \leftarrow (\mathcal{E}^*, \mathcal{V})$

3. *for* $k = K$ *to* 1

   (a) $\mathcal{E}_\times = \{e \in G_k \mid e \ltimes e' = \textit{true for some } e' \in G^* \setminus G_k\}$

   (b) $\mathcal{E}^* \leftarrow \mathcal{E}^* \setminus \mathcal{E}_\times$

   (c) $G^* \leftarrow (\mathcal{E}^*, \mathcal{V})$

   (d) $\mathcal{E}_\circ = \{e \in G^* \cap G_k \mid e \ltimes e' = \textit{true for some } e' \in G^* \cap G_k\}$

   (e) *if* $|\mathcal{E}_\circ| = 2$ *then*

      i. $\{e_1, e_2\} \leftarrow \mathcal{E}_\circ$

      ii. $\mathcal{E}^* \leftarrow \mathcal{E}^* \setminus e_1$

      iii. $G^* \leftarrow (\mathcal{E}^*, \mathcal{V})$

   (f) *end*

4. *end*

The algorithm can be explained as follows. In each iteration k of the loop, first get the edges in the $G_k \simeq \mathcal{U}_{\sigma_k}$ that intersect any edge of the remaining graph $G^* \setminus G_k$ (Step 3.a). These are all crossing edges in G_k that need not intersect any other edge in G_k. Remove these edges from $\mathcal{E}^*$ and update G^* (Step 3.c). Now find any remaining crossing edges in $G^* \cap G_k$ (Step 3.d). Any edges found in this step would be a subset of the crossing edges, out of which the crossing generator was originally constructed by Algorithm 6.2.1 or by (22). In fact, only two possibilities can occur. Either both crossing edges are removed in Step 3.b if they both intersected with some other edges in $G^* \setminus G_k$, so that $|\mathcal{E}_\circ| = 0$. Or, both edges are obtained if they only cross each other so that $|\mathcal{E}_\circ| = 2$. The case $|\mathcal{E}_\circ| = 1$ never happens because it needs 2 edges to obtain a crossing. When $|\mathcal{E}_\circ| = 2$, removing either of the two edges is equivalent and we follow the convention of removing the first edge in the set (Step 3.e.ii). The graph G^* is again updated. After k iterations of examining the crossing generators in the backward direction, we obtain $G^* = (\mathcal{E}^*, \mathcal{V})$. We will later denote it by G^*_Σ when the isomorphism G_Σ is considered. By construction, the algorithm fulfills all conditions for maximality as enumerated above. Therefore the graph $G^* = (\mathcal{E}^*, \mathcal{V})$ obtained at the end of the algorithm is indeed the maximal simplicial subgraph of G. ∎

We now give our main result.

Theorem 8.2.1 *A maximal simplicial subgraph G^* of a connectivity graph $G \in \mathcal{G}_{N,\delta}$ is given by the union,*

$$G^* = \left(\bigcup_i G^*_{\Sigma_i} \right) \cup H_\Delta.$$

Proof: By Proposition 6.2.2 any crossing edge of a connectivity graph is contained in some atomic crossing graph G_{Σ_i} generated by repeated Δ-amalgamations. Also by the same result, the intersection of the images of any two atomic crossing graphs G_{Σ_i} and G_{Σ_j} is made up of only smooth points (except the vertices). This means that the intersection of the two graphs has only non-crossing edges in common and the removal of crossing edges in one atomic crossing graph does not effect the crossing edges in the other graph. If we now obtain the maximal simplicial subgraph $G^*_{\Sigma_i}$ of each atomic crossing graph G_{Σ_i}, it does not result in the removal of any non-crossing edge in $G_{\Sigma_i} \cap G_{\Sigma_j}$. Therefore, if we obtain the maximal simplicial subgraph of each atomic crossing graph by executing Algorithm 8.2.1, then their union $\cup_i G^*_{\Sigma_i}$ is also maximal. Finally, the subgraph H_Δ contains no crossing edges and is already a maximal simplicial subgraph, thereby making

$$G^* = \left(\bigcup_i G^*_{\Sigma_i} \right) \cup H_\Delta,$$

a maximal simplicial subgraph of G. ∎

While the maximality condition captures the maximal simplicial structure in the connectivity graph, we can further see that any 3 vertices that span a triangle in G_s, can span a 2-simplex in the image. Therefore we have the following.

Definition 8.2.1 (Maximal Simplicial Complex) *$M_{G^*} = \{\mathcal{V}, \mathcal{S}\}$ is the maximal simplicial complex spanned by a connectivity graph if:*

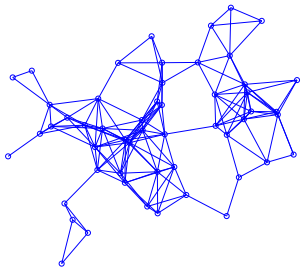
- *$\mathcal{S}_1 \subseteq \mathcal{S}$, where $\{\mathcal{V}, \mathcal{S}_1\}$ is the 1-simplicial complex of the maximal simplicial subgraph $G_* \subset G$.*
- *If a set of any three vertices $L = \{v_i, v_j, v_k\}$ form a cycle in $\{\mathcal{V}, \mathcal{S}_1\}$ then $L \in \mathcal{S}$*

The set of vertices $S = \{v_i, v_j, v_k\}$, that form a cycle in G_* induces a 2-simplex in K. Therefore M_{G_*} is a 2-complex made up of:

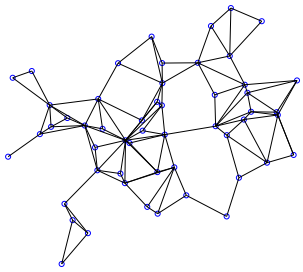
1. 1-simplexes induced by the edges of the graph

2. 2-simplexes induced by the cycles of 3 vertices of the graph

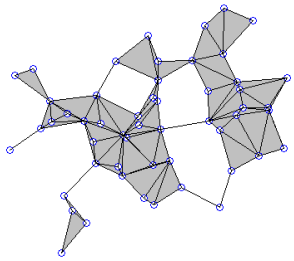
It may happen that M_{G_*} has no 2-simplexes. In this case the 1-skeleton is the complex itself case either. M_{G^*} is in fact the object associated with the topological shape of the formation. Some comments are appropriate at this point to explain why we have associated the topological characterization of the formation with the maximal simplicial complex spanned by its connectivity graph. Compare the lower connectivity graph on 3 vertices in Figure 5 with the graph in Figure 6.b. Both have a cyclic ring-like structure, which apparently makes them topologically equivalent. However, there is a subtle difference between the two, if we also desire that this structure is a decentralized multi-agent system. In the former example, all nodes interact directly with each other, whereas in the latter any node interacts directly with its two adjacent nodes only. Therefore the absence of an edge between opposite nodes creates a "hole" in the topological shape. By the method described in Definition 8.2.1, there would be a 2-simplex attached to the image of the graph of Figure 5 to get its maximal simplicial complex, making it a topological object of genus 0. On the other hand the maximal simplicial complex of the graph in Figure 6.b. would still have a hole (genus 1). Definition 8.2.1 lets us expand this point of view to more complex graphs. For example, the graph in Figure 24.a has 3 holes, each of which is bounded by 4 nodes or more, as seen in its maximal simplicial subgraph in Figure 24.b and its maximal simplicial complex in Figure 24.c. Once we have the simplicial complex of a graph, it can be studied using tools from standard algebraic topology to obtain its genus, fundamental groups, homological groups, etc. The process of ignoring the crossing edges corresponds to ignoring simplexes of higher dimension in the connectivity graph, which simplifies the computations. Furthermore, the characterization of the topological shape in terms of "holes" gives an indication of the absence or presence of coverage in an area enclosed by a network of spatially distributed sensors. Therefore, the connectivity graph and its maximal simplicial complex



(a). A connectivity graph G.



(b). The maximal simplicial subgraph G^*.



(c). The maximal simplicial complex M_{G^*}.

Figure 24: Decentralized triangulation of a connectivity graph.

are computationally cheap tools to study the topology of the corresponding Rips-complex. However, before we give this type of topological characterization to connectivity graphs, we should investigate the issue of uniqueness for such characterization.

8.3 Uniqueness of Topological Characterization

It should be noted that maximal simplicial subgraph of a connectivity graph (and therefore the maximal simplicial complex) need not be unique and depends on the order in which crossing edges are removed, i.e. the order in which Δ-amalgamations are repeated to obtain each atomic crossing graph. For example, consider the situation in Figure 15. Starting with the connectivity graph G shown there, we can obtain a decomposition of the graph G using

atomic crossing generators by executing Algorithm 6.2.1. It can be seen that A maximal simplicial subgraph can be obtained by executing Algorithm 8.2.1. The resulting graph G_1^* is depicted in Figure 25.a. However, upon close examination we can see that the subgraph G_2^* shown in Figure 25.b is also maximal. This should not come as a surprise, as different maximal simplicial graphs emerge as a result of the removal of crossing edges in different order. Now, consider the maximal simplicial complexes associated with the two maximal simplicial subgraphs, as shown in Figures 25.c and 25.d. It is obvious from the figures that the two simplicial complexes differ in their homology groups. The maximal simplicial complex of the subgraph obtained as a result of executing Algorithm 8.2.1 has genus 1, whereas the one depicted in Figure 25.d has a trivial fundamental group and hence genus 0. [1] Therefore, the choice exercised in removal of crossing edges makes a difference in the topological characterization of the connectivity graph. As explained above, the presence or absence of a "hole" in the simplicial complex is an indicator of presence or absence of direct interaction between nodes of a connectivity graph. Since we are interested in obtaining a topological characterization that truly captures the lack or presence of direct interactions, we must modify Algorithms 6.2.1 and 8.2.1. For obtaining such a modification, we define a quotient space on Δ-amalgamations as follows.

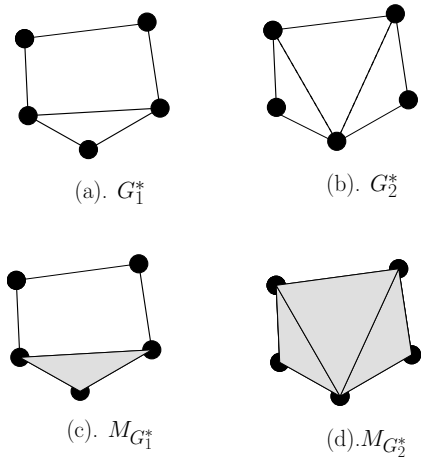


Figure 25: Two maximal simplicial complexes of the same connectivity graph

We first define the following equivalence relation. Let $G_1 *_\Delta G_2$ and $H_1 *_\Delta H_2$ be two

[1] For details on how to calculate fundamental groups of simplicial complexes, see [5, 49].

well-defined Δ-amalgamations, then $G_1 *_\Delta G_2 \sim H_1 *_\Delta H_2$ if and only if:

1. $G_i \simeq H_i \simeq \mathcal{U}_{\sigma_i}$ for some $\sigma_i \in \{1, 2, 3\}$, and for $i = 1, 2$.
2. $G_1 *_\Delta G_2 \simeq H_1 *_\Delta H_2 \simeq \mathcal{U}_{\sigma_1} *_\Delta \mathcal{U}_{\sigma_2}$.
3. Number of non-smooth points in $I_{G_1 *_\Delta G_2}$ is equal to the number of non-smooth points in $I_{H_1 *_\Delta H_2}$.

The members of an equivalence class under this equivalence relation is denoted by $[G_1 *_\Delta G_2]$. We also define a quotient map π that sends $G_1 *_\Delta G_2$ to $[G_1 *_\Delta G_2]$. If we denote the space of all well-defined Δ-amalgamations between crossing generators as $\mathcal{U} *_\Delta \mathcal{U}$, then the resulting quotient space can be denoted by $\mathcal{U} *_\Delta \mathcal{U}/_\sim$. By enumeration techniques it can be seen that the quotient space is composed of only 7 equivalence classes, as shown in figure Figure 26. The details of this enumeration are omitted here for brevity.

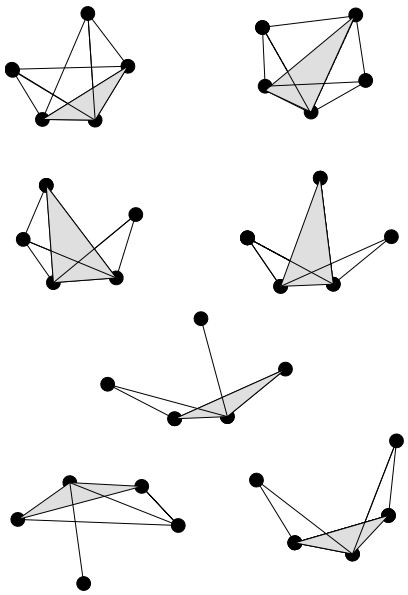


Figure 26: Members of the quotient space $\mathcal{U} *_\Delta \mathcal{U}/_\sim$

Now let us define a map $\varphi : \mathcal{U} *_\Delta \mathcal{U}/_\sim \longrightarrow \mathcal{G}_5$ as depicted in Figures 27. This map is used to remove crossing edges whenever the crossing edges of a connectivity graph are captured as atomic crossing graphs. Since, π is also a graph isomorphism, it makes sense to consider the map $\pi^{-1}|_{\varphi(.)}$ which translates the edge removal back to the original graph. We have the following commutative diagram.

$$\begin{array}{ccc} \mathcal{U} *_\Delta \mathcal{U} & \xrightarrow{\varphi_*} & \mathcal{G}_5 \\ \Big\downarrow \pi & & \Big\uparrow \pi^{-1} \\ \mathcal{U} *_\Delta \mathcal{U}/_\sim & \xrightarrow{\varphi} & \mathcal{G}_5 \end{array}$$

Thus, if $G_1, G_2 \subset G$, and $G_1 *_\Delta G_2$ is well-defined then, we can obtain a maximal simplicial subgraph of $\overline{G_1 *_\Delta G_2}$ using the map $\varphi_* = \pi^{-1} \circ \varphi \circ \pi$, i.e. $\varphi_*(G_1 *_\Delta G_2)$ is a maximal simplicial subgraph of $\overline{G_1 *_\Delta G_2}$. For repeated well-defined Δ-amalgamations, of the type given in (22), the Algorithm 8.2.1 can be modified as follows.

Algorithm 8.3.1 .

1. $G^* \leftarrow \emptyset$
2. *for* $k = K$ *to* 2
   (a) $G^* \leftarrow G^* \cup \varphi_*(G_{i_{k-1}} *_\Delta G_{i_k})$
3. *end*

Therefore the maximal simplicial subgraph of $\overline{G_{i_1} *_\Delta G_{i_2} *_\Delta \ldots G_{i_k}} \in \mathcal{G}_{4+k,\delta}$ can be obtained by

$$\bigcup_{j=1}^{k-1} \varphi_*(G_{i_j} *_\Delta G_{i_{j+1}})$$

The maximal simplicial subgraph obtained in this way is free of the inconsistencies explained earlier. One can now proceed to compute all sorts of topological invariants associated with the simplicial complex.

The geometric structure and topological characterization of the connectivity graphs of multi-agent formations given in this work can be used to obtain certain global properties of formations using decentralized algorithms, suitable for implementation on scalable, totally decentralized multi-agent systems. It will be appropriate to mention, that the entire machinery presented here for decomposing connectivity graphs into simplicial complexes becomes irrelevant, if the computations are performed in a centralized manner. The centralized method of obtaining the simplicial subgraph can be much simpler and need not

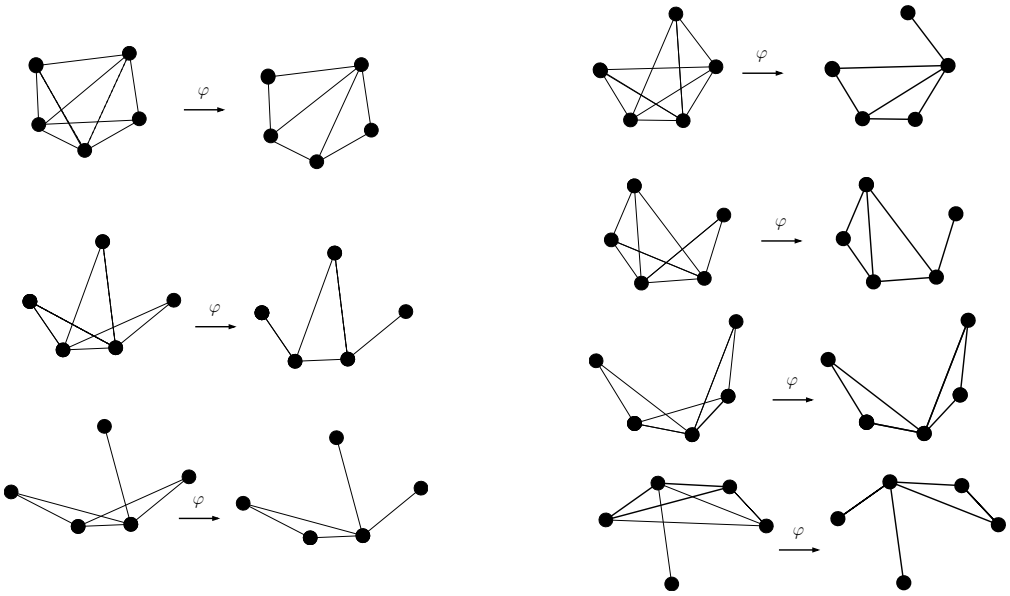


Figure 27: Results of the crossing-edge removal map φ on members of $\mathcal{U} *_{\Delta} \mathcal{U}/_{\sim}$.

have knowledge of the local geometrical structure obtained above. The maximal simplicial subgraph G^* of a connectivity graph $G \in \mathcal{G}_{N,\delta}$ can therefore be obtained using a decentralized algorithm. The detection of genus is also implementable as a decentralized algorithm. Since the fundamental group $\pi_1(I_{M_{G^*}})$ is isomorphic to the edge group $E(M_{G^*})$ of the triangulation, we can base our method on the reduction of loops inside M_{G^*}, according to the equivalence rules of $E(M_{G^*})$ [5]. Using this approach, one can implement a decentralized algorithm to find out if M_{G^*} has genus 0 or not. The role of low-complexity formations called δ-chains (Hamiltonian paths) has moreover been studied and emphasized in a previous chapter. Decentralized algorithms for obtaining these δ-chains are of considerable interest, and will be explained in later chapters.

8.4 Open Problem III: Decentralized Computation of Homology Groups

Perhaps the most crucial direction for this work is to develop a local criterion from which a distributed algorithm could be developed for the computation of homology groups. This is particularly important for the coverage criterion developed in Chapter 11. This seems theoretically possible, since homology is a local operation. Based on some preliminary

studies it is anticipated that this can be done using the technique of spectral sequences from homological algebra. However, coming up with a practical algorithm for achieving that is a significant challenge.

PART II

Dynamic Aspects: Connectivity

Graph Processes

CHAPTER IX

FEASIBLE AND REACHABLE SETS IN CONNECTIVITY GRAPH PROCESSES

As mentioned in the introduction, formation switching with limited global information is an important problem in multi-agent robotics. However, little work has been done so far that adequately addresses the problem of formation switching under limited range constraints. Therefore the ability to give exact certificates about what can and cannot be achieved under these constraints is a desirable result. Recall, that the connectivity graph of the formation evolves in time as $G(t) = (\mathcal{V}, \mathcal{E}(t))\Phi_N(\mathbf{x}(t))$. Under standard assumptions on the individual trajectories of agents, one gets a finite sequence of graphs $\{G_0, G_1, \ldots, G_M\}$ for each finite interval of evolution $[0, T]$, where $\Phi_N(\mathbf{x}(0)) = G_0$, G_i switches to G_{i+1} at time t_i, and $\Phi_N(\mathbf{x}(T)) = G_M$. We will often write this as $G_0 \to G_1 \to \ldots \to G_M$ and call such a sequence a *connectivity graph process*[1]. Such graph processes can be thought of as trajectories on the space $\mathcal{G}_{N,\delta}$. An example of such graph processes has been illustrated in Figure 28. In what follows we discuss the role of *feasible, reachable* and *desirable* sets when generating trajectories on this space.

9.1 Feasible Connectivity Graph Transitions

The connectivity graph processes are generated through through the movement of individual nodes. For a connectivity graph $G_j(\mathcal{V}_j, \mathcal{E}_j) = \Phi_N(\mathbf{x}(t_j))$ let the nodes be partitioned as $\mathcal{V}_j = \mathcal{V}_j^0 \cup \mathcal{V}_j^m$, where the movement of the nodes in V_j^m facilitates the transition from G_j to the next graph G_{j+1} and V_j^0 is the set of nodes that are stationary. With the positions $\mathbf{x}_j^0 = \{\mathbf{x}_m(t_i)\}_{m \in \mathcal{V}_j^0}$ being fixed, let $\text{Feas}(G_j, \mathcal{V}_j^m, \mathbf{x}_j^0) \subseteq \mathcal{G}_{N,\delta}$ be the set of all connected connectivity graphs that are feasible by an unconstrained placement of positions corresponding to $\mathcal{V}_j^m$ in $\mathbb{R}^2$. (We will often denote this set as $\text{Feas}(G_j, \mathcal{V}_j^m)$, when the

[1] We borrow this term from Mesbahi [71]

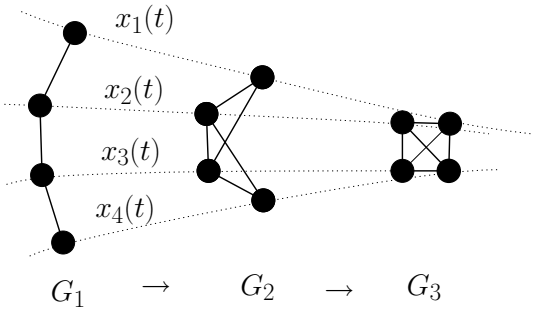


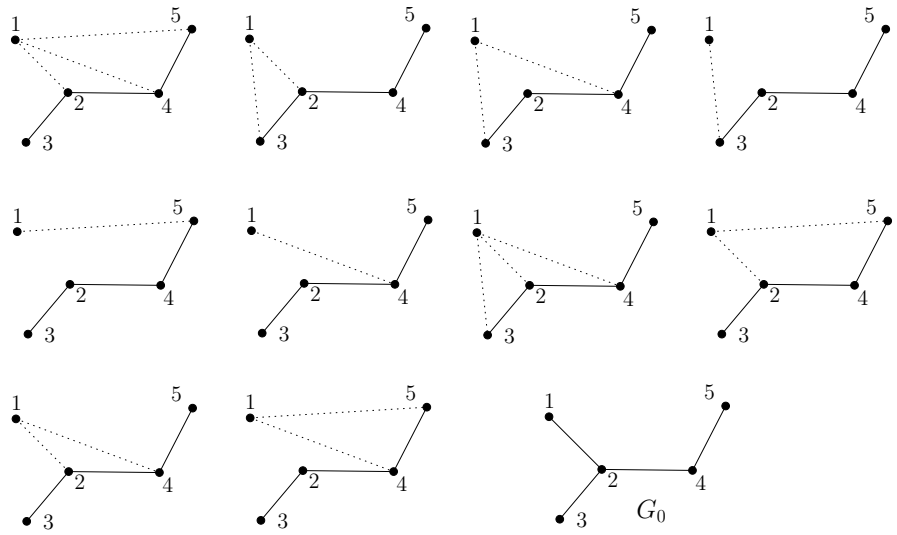
Figure 28: A connectivity graph process as a result of movements of individual nodes.

the positions $\mathbf{x}_j^0$ are understood from context.) The set $\mathrm{Feas}(G_j, \mathcal{V}_j^m)$ of feasible graph transitions can be computed using the semi-definite programming methods discussed above. It will be appropriate to explain the reason for keeping track of mobile and stationary nodes at each transition. In principle, it is possible to compute this entire set by an enumeration procedure. However, in order to manage the combinatorial growth in the number of possible graphs, it is desirable to let the transitions be generated by the movements of a small subset of nodes only. In fact, we will investigate the situation where only one node is allowed to move at a time. Let $\mathcal{V}_j^0 = \{1, \ldots, k-1, k+1, \ldots, N\}$ and $\mathcal{V}_j^m\{k\}$. It should be noted that the movement of node k can only result in the addition or deletion of edges that have node k as one of its vertices. Therefore the enumeration of the possible resulting graphs should count all possible combinations of such deletions and additions. This number can be easily seen to be 2^{N-1} for N nodes. Since we are also required to keep the graph connected at all times, this number is actually $2^{N-1} - 1$, obtained after removing the graph in which node k has no edge with any other node.

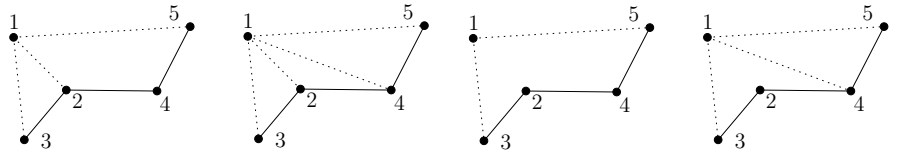
Now, we can use the S-procedure to evaluate whether each of the new graphs resulting from this enumeration is feasible. Since all nodes are fixed except for $\mathbf{x}_k = (x, y)$, the semi-algebraic set we need to check for non-feasibility is defined by $N-1$ polynomial inequalities over $\mathbb{R}[x, y]$. Each of these inequalities has either of the following two forms,

$$\delta^2 - (x - x_i)^2 - (y - y_i)^2 \geq 0, \text{ if } e_{ki} \in \mathcal{E},$$
$$(x - x_i)^2 + (y - y_i)^2 - \delta^2 > 0, \text{ if } e_{ki} \notin \mathcal{E}.$$

where $2 \leq i \leq N$, $\mathcal{E}$ is the edge set of the new graph and we denote $\mathbf{x}_i(t_j)$ by (x_i, y_i)



Set of feasible transitions $Feas(G_0, \{1\})$.



Infeasible transitions.

Figure 29: Feasible and infeasible graphs by movement of node 1.

for $i \neq k$. This computation can be repeated for all N nodes so that we have a choice of $N(2^{N-1}-1)$ graphs. Each of the $N-1$ inequalities can be written as either

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & x_i \\ 0 & -1 & y_i \\ x_i & y_i & \delta^2 - x_i^2 - y_i^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \geq 0, \text{ if } e_{ik} \in \mathcal{E},$$

or

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ -x_i & -y_i & x_i^2 + y_i^2 - \delta^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} > 0, \text{ if } e_{ik} \notin \mathcal{E}.$$

Denoting by

$$A_i = \begin{bmatrix} -1 & 0 & x_i \\ 0 & -1 & y_i \\ x_i & y_i & \delta^2 - x_i^2 - y_i^2 \end{bmatrix}, \qquad B_i = \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ -x_i & -y_i & x_i^2 + y_i^2 - \delta^2 \end{bmatrix},$$

and ignoring the lossy aspect of the S-procedure [17], we need to solve the LMI,

$$-A_{\alpha_1} - \sum_{i \neq 1, e_{\alpha_i k} \in \mathcal{E}} \lambda_{\alpha_i} A_{\alpha_i} - \sum_{j, e_{\alpha_j k} \notin \mathcal{E}} \lambda_{\alpha_j} B_{\alpha_j} \geq 0.$$

An example of such a calculation is given in Figure 29, where $\mathcal{V}^0 = \{2, 3, 4, 5\}$ and $\mathcal{V}^m\{1\}$. The LMI control toolbox [68] for MATLAB has been used to solve the LMI for each of these graphs in order to get the appropriate certificates.

9.2 Reachability in Connectivity Graph Processes

Note that the set $\text{Feas}(G_0, \mathcal{V}_0^m)$ does not depend on the actual movement of the individual nodes. In fact, even if $G \in \text{Feas}(G_0, \mathcal{V}_0^m)$, it does not necessarily mean that there exists a trajectory by which $G_0 \to G$ or even that $G_0 \to G_f \ldots G$. The geometrical configuration of nodes may create an obstruction in obtaining a graph process that takes G_0 to G. There are two ways by which this obstruction is created: The requirement to maintain connectivity and conformity to a fixed set of mobile nodes. We therefore need some notion of *reachability* on the space $\mathcal{G}_{N,\delta}$. We say that a connectivity graph G_f is *reachable* from an initial graph G_0 if there exists a connectivity graph process of finite length $G_0 \to G_1 \to \ldots G_f$ and a sequence of vertex-sets $\{\mathcal{V}_k^m\}$ such that each $G_{k+1} \in \text{Feas}(\mathcal{G}_k, \mathcal{V}_k^m)$. If $\mathcal{V}_k^m = \mathcal{V}^m$ at each transition, then every $G_k \in \text{Feas}(G_0, \mathcal{V}^m)$. (In particular, $G_f \in \text{Feas}(G_0, \mathcal{V}^m)$.) Consider all such G's that are reachable from G_0 with a fixed $\mathcal{V}_m$. We will denote this set by $\text{Reach}(G_0, \mathcal{V}^m)$. It is easy to see that $\text{Reach}(G_0, \mathcal{V}^m) \subseteq \text{Feas}(G_0, \mathcal{V}^m)$.

In the previous section, it was shown how to determine the membership for the set $\text{Feas}(G_0, \mathcal{V}^m)$. For all such graphs, determining whether they also belong $\text{Reach}(G_0, \mathcal{V}^m)$ is not very straightforward, specially under the restriction that the intermediate graphs have to be connected. For the special case of a single mobile node, the situation is manageable as discussed in the next chapter.

CHAPTER X

CYLINDRICAL ALGEBRAIC DECOMPOSITION AND PATH PLANNING

In this chapter we describe a method of computing reachable sets in connectivity graph processes. We use a technique in real algebraic geometry known as the cylindrical algebraic decomposition (CAD) [10]. This decomposition not only lets us compute the reachable sets, but also provides an explicit planning algorithm for generating the connectivity graph processes that take an initial graph to a final graph in the reachable set.

10.1 Cylindrical Algebraic Decomposition Algorithm

We first give some definitions. A *semialgebraic set* in $\mathbb{R}^d$ is a subset of $\mathbb{R}^d$, which is a finite Boolean combination of sets of the form $\{\mathbf{x} \in \mathbb{R}^d \mid f(\mathbf{x}) > 0\}$ and $\{\mathbf{x} \in \mathbb{R}^d \mid g(\mathbf{x}) = 0\}$, where $f, g \in \mathbb{R}[x_1, \cdots, x_d]$. A *Nash manifold* $M \in \mathbb{R}^n$ is an analytic submanifold which is a semialgebraic set. Let $U \subset \mathbb{R}^n$. A *Nash function* $f : U \to \mathbb{R}$ is a smooth function for which there exists a polynomial $p(x,t) = p(x_1, \ldots x_n, t)$ such that $p(x, f(x)) = 0$ for all $x \in U$. A *Nash cell* in $\mathbb{R}^n$ is a Nash manifold which is diffeomorphic to an open box $(-1,1)^d$ of dimension d. Every semialgebraic set can be decomposed into a disjoint union of Nash cells. More precisely [10, 15];

Theorem 10.1.1 *Let $A_1, \cdots, A_q$ be semialgebraic subsets of $\mathbb{R}^n$. Then there exists a finite semialgebraic partition of $\mathbb{R}^n$ into Nash cells such that each A_j is a union of some of these cell.*

The existence of such a decomposition is given by a technique in real algebraic geometry known as the cylindrical algebraic decomposition (CAD) [10]. A CAD of $\mathbb{R}^n$ is a partition into finitely many semialgebraic subsets. The CAD of $\mathbb{R}^n$ is given by induction on n as follows.

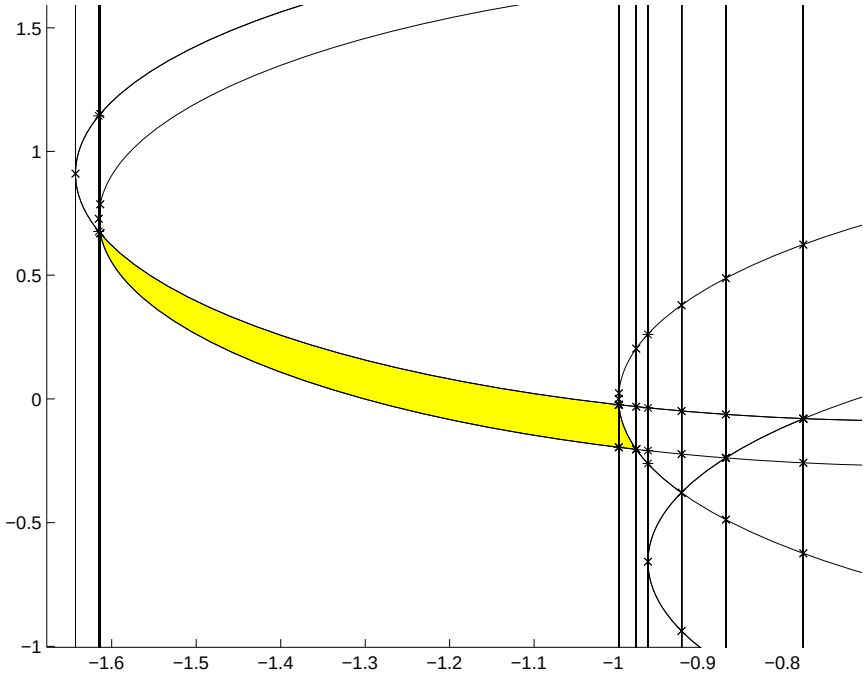


Figure 30: A CAD of $\mathbb{R}^2$.

1. A CAD of $\mathbb{R}$ is a subdivision by finitely many points $a_1 < \ldots < a_l$. The cells are the singletons $\{a_i\}$ and the open intervals delimited by these points.

2. For $n > 1$, a CAD of $\mathbb{R}^n$ is given by a CAD of $\mathbb{R}^{n-1}$and for each cell C of $\mathbb{R}^{n-1}$, the Nash functions $\zeta_{C,1} < \cdots < \zeta_{C,l_C} : C \to \mathbb{R}$.

The cells of the CAD of $\mathbb{R}^n$ are the graphs of $\zeta_{C,j}$ and the bands in the cylinders $C \times \mathbb{R}$ determined by the graphs. Observe that the cells generated by CAD are Nash, thus providing a partition satisfying Theorem 10.1.1. For our need, we only need a CAD of $\mathbb{R}^2$. One such decomposition is shown in Figure 30. The 0-cells are marked by crosses. The 1-cells are the curves and vertical lines joining the 0-cells. The 2-cells are the regions bounded by the 1-cells. One 2-cell is shaded for illustration. For a more applied introduction to CAD we refer the interested reader to [61]. For notational convenience, let us denote the collection of Nash cells corresponding to a semialgebraic set S by CAD(S).

10.2 Path Planning Using CAD

We now perform this decomposition for the following special case. With $\mathcal{V}^m = \{k\}$ and $\mathcal{V}^0 = \{1, \cdots, k-1, k+1, \cdots, N\}$, consider the semialgebraic set

$$X(\mathcal{V}^0) = \bigcup_{j \in \mathcal{V}^0} \{(x, y) : (x - x_j)^2 + (y - y_j)^2 \leq \delta^2\} \subset \mathbb{R}^2. \tag{28}$$

The first thing to note about this set is that it is compact. This relieves us of any complicated compactification procedures that are needed to get the CAD of non-compact sets. Let C be a cell in $\mathrm{CAD}(X(\mathcal{V}^0))$, then we call $r \in C$ a *configuration* point in C if $r \neq \mathbf{x}_j$ for all $j \in \mathcal{V}^0$. We then have the following result.

Proposition 10.2.1 *Let $C \in \mathrm{CAD}(X(\mathcal{V}^0))$. If p, q are any two configuration points in C then $\Phi_N((\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, p, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_N)) = \Phi_N((\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, q, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_N))$.*

Proof: If C is a 0-cell we have the result trivially. We therefore assume that $\dim(C) > 0$. For notational convenience denote $(\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, p, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_N)$ by $\mathbf{x}_p$ and the graph $\Phi_N((\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, p, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_N))$ by G_p for a point $p \in C$.

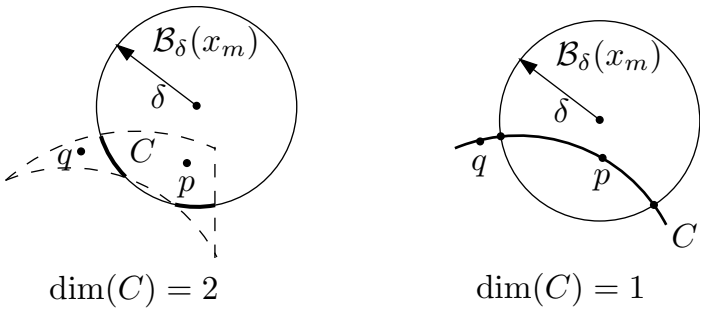


Figure 31: Topological obstruction in the proof of Proposition 10.2.1.

First note that any pair of positions $\mathbf{x}_i, \mathbf{x}_j$, where $i, j \in \mathcal{V}^0$ are always unequal. This is because they are the product of a valid configuration on the configuration space $\mathcal{C}^N(\mathbb{R}^2)$ described in Equation 5. If p is a configuration point then the N-tuple $\mathbf{x}_p = (\mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, p, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_N)$ is a valid configuration on $\mathcal{C}^N(\mathbb{R}^2)$, hence the name configuration point. The configuration point p would correspond to the k-th node in G_p. This shows that the mapping $\Phi_N(\mathbf{x}_p)$ is well defined for all configuration points in C. We now prove the proposition by contradiction.

Assume that there exist points $p, q \in C$ such that $G_p \neq G_q$. The graph G_p would differ from G_q in at least one edge incident on node k. Without loss of generality, assume that e_{km} is an edge between a node $m \in \mathcal{V}^0$ and the k-th node corresponding to p in G_p, and that there is no edge between nodes k and m in G_q. The existence of this edge in G_p means that p is inside the closed ball $\mathcal{B}_\delta(\mathbf{x}_m)$, while $q \notin \mathcal{B}_\delta(\mathbf{x}_m)$ as shown in Figure 31. From the definition of a Nash cell, C is diffeomorphic to $(-1,1)^{\dim(C)}$ which is simply connected.[1] This means that C, in addition to being an open set, is also simply connected. Therefore the ball $\mathcal{B}_\delta(\mathbf{x}_m)$ induces a partition

$$C = (C \setminus \mathcal{B}_\delta(\mathbf{x}_m)) \cup (C \cap \mathcal{B}_\delta(\mathbf{x}_m)),$$

such that $\partial\mathcal{B}_\delta(\mathbf{x}_m)) \cap C \neq \emptyset$. From Figure 31, it is clear that each connected component of $\partial\mathcal{B}_\delta(\mathbf{x}_m)) \cap C$ induces $\dim(C) - 1$ Nash cells that are not already present in the CAD of $\mathrm{CAD}(X(\mathcal{V}^0))$. Moreover, this means that there exist lower dimensional Nash cells that are properly contained in C. Both implications are contrary to the CAD construction described above. Therefore, the connectedness of C creates a topological obstruction in getting $G_p \neq G_q$. This proves the Proposition. ∎

From this we get the following useful result.

Corollary 10.2.1 *Let $\mathbf{x}(t) = (\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \gamma(t), \mathbf{x}_{k+1}, \ldots, \mathbf{x}_N)$ be a trajectory on $\mathcal{C}^N(\mathbb{R}^2)$. If $\gamma(t) \in C$ for all $t \in [t_0, t_f)$ then the connectivity graph process has no transitions while $t \in (t_0, t_f)$.*

For a trajectory $\mathbf{x}(t) = (\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}, \gamma(t), \mathbf{x}_{k+1}, \ldots, \mathbf{x}_N)$ on $\mathcal{C}^N(\mathbb{R}^2)$ that is not confined to a single Nash cell, the graph may change as the trajectories goes from one cell to another. Moreover, since the connectivity graph remains unchanged inside one cell, the transition must take place at the boundary of Nash cells. We can now begin to appreciate the connection between the CAD decomposition and the geometric origin of transitions in a connectivity graph process. We further observe that the trajectory is partitioned into a *finite* number of pieces, where each piece is confined to a single Nash cell. In more precise terms,

[1] Topologically, a simply connected set means path-wise connectedness and the absence of any *holes* in the set.

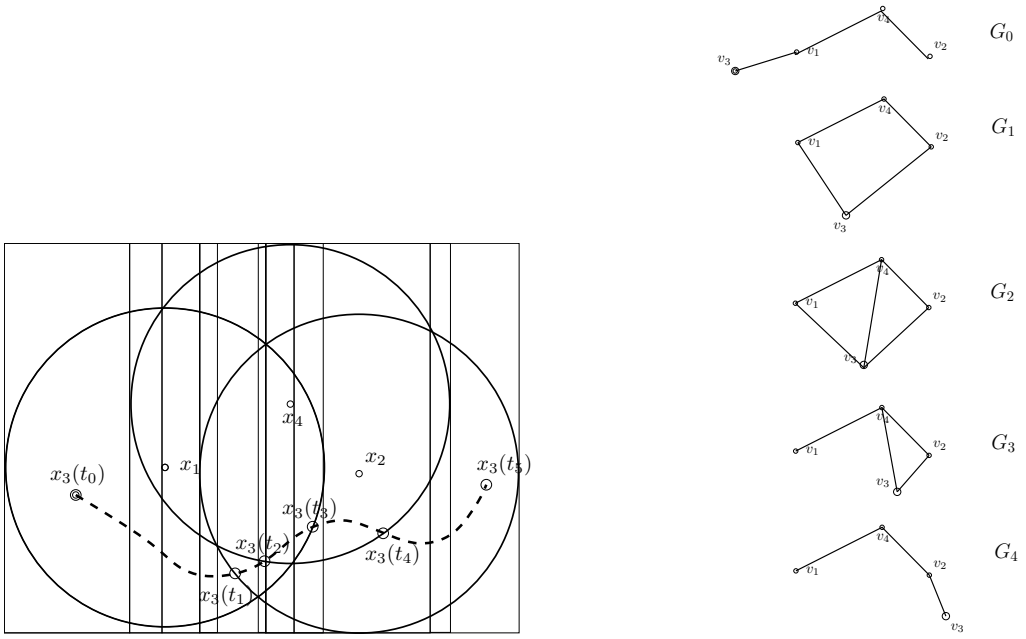


Figure 32: A trajectory and its connectivity graph process.

for each graph process $G_0 \to G_1 \ldots \to G_N$ with $\mathcal{V}_i^m = \{k\}$ for $0 \leq i \leq N$ and transitions at $t_1 < t_2 < \ldots < t_N$, there exist a finite number of Nash cells $\mathcal{C}_{\mathbf{x}} = \{C_0, C_1, \ldots, C_M\} \subseteq \mathcal{C}$, where $M \geq N$, such that the trajectory $\mathbf{x}(t)$ intersects with a finite sub-collection $\mathcal{C}_{\mathbf{x},i} \subseteq \mathcal{C}_{\mathbf{x}}$ for $t \in (t_i, t_{i+1})$. One such trajectory and the corresponding graph process has been depicted in Figure 32. We will construct an explicit planning algorithm to go from one point in the CAD to another. Let us define the k-th skeleton of the CAD of a set S as

$$\mathrm{CAD}^{(k)}(S) = \{C \in \mathrm{CAD}(S) \mid \dim(C) = k\}.$$

In particular, the 1-skeleton $\mathrm{CAD}^{(1)}(S)$ consists of all 1-dimensional Nash cells in $CAD(S)$. Recall that the set $X(\mathcal{V}^0)$ described by Equation 28 is a union of closed disks in $\mathbb{R}^2$. Moreover the boundary of of this set

$$\partial X(\mathcal{V}^0) \subseteq \mathrm{CAD}^{(0)}(X(\mathcal{V}^0)) \cup \mathrm{CAD}^{(1)}(X(\mathcal{V}^0)).$$

If $C^2 \in \mathrm{CAD}^{(2)}(S)$ and $\bar{C^2}$ denotes its closure in $\mathbb{R}^2$, then by construction of the CAD, $\partial \bar{C^2} \subseteq \mathrm{CAD}^{(0)}(X(\mathcal{V}^0)) \cup \mathrm{CAD}^{(1)}(X(\mathcal{V}^0))$. Therefore, any connected component of $X(\mathcal{V}^0)$ cannot be made up of cells in $\mathrm{CAD}^{(2)}(S)$ alone. In fact, we get the following useful lemma.

Lemma 10.2.1 *The set* $\mathrm{CAD}(X(\mathcal{V}^0))$ *is connected if and only if* $\mathrm{CAD}^{(0)}(X(\mathcal{V}^0)) \cup \mathrm{CAD}^{(1)}(X(\mathcal{V}^0))$ *is connected.*

The above lemma suggests the existence of a path planning algorithm for moving the mobile node between any two points of a connected component of $X(\mathcal{V}^0)$. Given p, q in the same connected component of $X(\mathcal{V}^0)$, we construct a path $\gamma : [0, T] \to X(\mathcal{V}^0)$ such that $\gamma(0) = p$ and $\gamma(T) = q$. The placement of both points p, q in the same connected component ensures that $T < \infty$. Also, each of the points p, q lie in their unique cells of $\mathrm{CAD}(X(\mathcal{V}^0))$, which we denote by C_p, C_q respectively. We build our algorithm from the following observations.

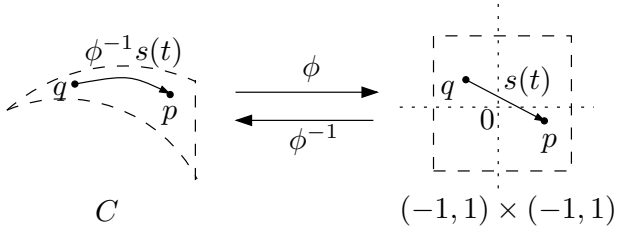


Figure 33: Path between two points p, q in a 2-cell of the CAD.

Case 1: When $C_p = C_q = C$, i.e. when both points lie in the same cell, consider the diffeomorphism $\phi : C \to (-1, 1)^{\dim(C)}$. Then the points $\phi(p)$ and $\phi(q)$ in the open cube can be joined by a straight line $s(t) = t\phi(p) + (1 - t)\phi(q)$. Now map this line back to C by

$$\gamma(t) = \phi^{-1}(s(t)),\ t \in [0, 1].$$

This gives us the desired path. Such an explicit construction in terms of the diffeomorphism may only be needed when $\dim(C) = 2$, as depicted in Figure 33. For lower dimensions the construction of $\gamma(t)$ is rather obvious.

Case 2: When $C_p \neq C_q$. We study various situations in this case.

(a) . When $\dim(C_p), \dim(C_q) < 2$. In this case one can construct the path γ explicitly by building a graph $\mathcal{G}'_P = (\mathcal{V}'_P, \mathcal{E}'_P)$ in the following way. Each $v_i \in \mathcal{V}'_p$ corresponds to a 0-cell C_i^0 in the $\mathrm{CAD}^{(0)}(X(\mathcal{V}^0))$ and an edge $e_{ij} \in \mathcal{E}'_p$ if and only if there

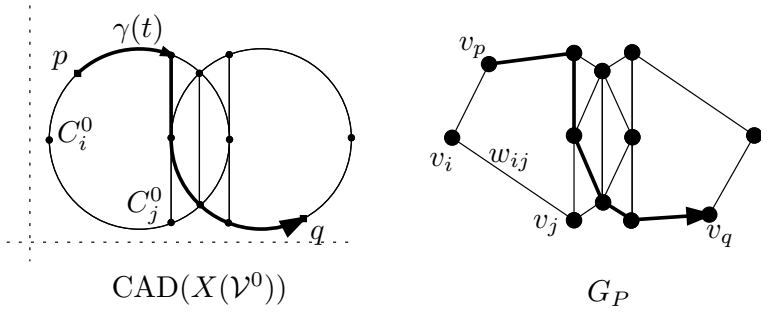


Figure 34: Path planning via planning graph.

exists a 1-cell $C^1 \in \mathrm{CAD}^{(1)}(X(\mathcal{V}^0))$ such that $C_i^0, C_j^0 \in \partial C^1$. Note that by construction of the CAD, if such a 1-cell exists, it will be unique. Moreover, each 1-cell will be mapped to a unique edge in $\mathcal{G}'_P$. If $\dim(C_p)\dim(C_q) = 0$ we are done. If not, we modify $\mathcal{G}'_P$ as follows. We add one vertex for each of the points p, q that belong to a 1-cell. Suppose $\dim(C_p)1$. Then C_p corresponds to an edge $e_{jk} = (v_j, v_k)$ induced by 0-cells C_j^0 and C_k^0. We add a vertex v_p corresponding to p. Now modify the graph $\mathcal{G}'_P$ by removing the edge e_{jk} and inserting two edges $e_{jp} = (v_j, v_p)$ and $e_{pk} = (v_p, v_k)$. A similar procedure modifies $\mathcal{G}'_P$ if $\dim(C_q) = 1$. We call this modified graph our planning graph G_P. An example of this construction is shown in Figure 34. We can now convert this graph into a *weighted* graph, by assigning each edge e_{ij} a real number w_{ij}. One choice of weights can be a constant weight on all edges. We will show later how to assign a more natural set of weights. Once we have that, we can use a standard discrete planning algorithm such as the *Dijkstra's algorithm* [30] to get a path on $\mathcal{G}_P$ that connects v_p to v_q by a sequence of edges and vertices. This graphical path can now be mapped back to $\mathbb{R}^2$ to get an explicit path $\gamma(t)$ that connects p to q. We now give this mapping.

Note first that a 1-cell C_j^1 has two 0-cells say $C_{j_0}^0$ and $C_{j_1}^0$ on its boundary. If this 1-cell is also a subset of $\partial\mathcal{B}_\delta(\mathbf{x}_k)$ for some $k \in \mathcal{V}^0$ then by the properties of CAD construction, C_j^1 has a natural parametrization as a curve in $\mathbb{R}^2$ given by

$$c_j : s \mapsto (s, \mathrm{sgn}(C_j^1)\sqrt{\delta^2 - (s - y_k)^2} + x_k), \quad s \in (x_{j_0}, x_{j_1}),$$

where $\text{sgn}(C_j^1)$ can be determined uniquely for each such 1-cell. If $w_{j_0j_1}$ is the weight on the corresponding edge $e_{j_0j_1} \in \mathcal{E}_P$, then we can re-parameterize by a linear mapping $T_j : \mathbb{R} \to \mathbb{R}$ that sends $[0, w_{j_0j_1}) \mapsto [x_{j_0}, x_{j_1})$.We can therefore define a parameterized curve $\gamma_{j_0j_1}(t) = c_j(T_j(t))$, where $t \in [0, w_{j_0j_1})$.

In case the 1-cell C_k^1 is a vertical line segment, the parametrization is more simple. Simply, let $\gamma_{k_0k_1}(t) = T_k(t)$ where $t \in [0, w_{k_0k_1})$ and T_k maps $[0, w_{k_0k_1}) \mapsto [y_{k_0}, y_{k_1})$. In this way we have a method to map back an edge in the planning graph to $\mathbb{R}^2$. We have omitted here details for edges that have either v_p or v_q as one of the vertices. It is easy to see that the mapping for these edges is very similar.

Now, given a path in $\mathcal{G}_P$ as a sequence of vertices and edges, one can build explicitly a path in $\mathbb{R}^2$ as a concatenation of the curves described above. More precisely let the path be given by a sequence $v_p, e_{pk_1}, v_{k_1}, e_{k_1k_2}, v_{k_2}, \ldots, v_{k_M}, e_{k_Mq}, v_q$, then we define a path that connects p to q by

$$\gamma(t) = \gamma_{pk_1} \circ \gamma_{k_1k_2} \circ \ldots \circ \gamma_{k_Mq}(t), \qquad t \in [0, W)$$

where $W = w_{pk_1} + w_{k_1k_2} + \ldots w_{k_Mq}$ and $\circ$ is the path concatenation operation defined by

$$\gamma_1 \circ \gamma_2(t) = \begin{cases} \gamma_1(t), & t \in [0, w_1); \\ \gamma_2(t), & t \in [w_1, w_1 + w2). \end{cases}$$

Finally, note that there is a natural way of assigning weights to $\mathcal{G}_P$ by letting each weight w_ij be equal to the length of the corresponding curve γ_{ij} in $\mathbb{R}^2$.

$$w_{ij} = \int_{\mathbf{x}_i}^{\mathbf{x}_j} \|\dot{\gamma}_{ij}\|,$$

This gives us constant speed paths in $\mathbb{R}^2$.

(b). General Case: We now discuss the most general case. We present a strategy for the case when either or both of p and q belong to a 2-cell. Suppose p, q are inside $C_p^2, C_q^2 \in \text{CAD}^{(2)}(X(\mathcal{V}^0))$ respectively. Now consider the line segment $\gamma_0(s) = sq + (1-s)p$, where $s \in [0, 1]$. By the construction of CAD, this line

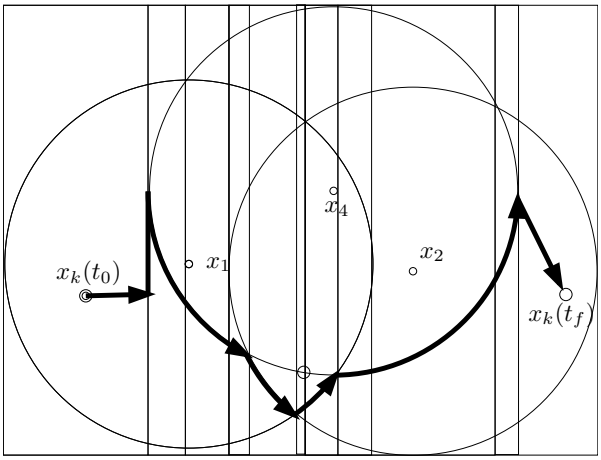


Figure 35: Path between two points belonging to their 2-cells in the CAD.

segment cuts at least 2 cells of dimension less than 2. Let s_1 be the minimum value of s at which the line cuts a lower dimensional cell. Also let s_2 be the corresponding maximum value. Now let $p' = \gamma_0(s_1)$ and $q'\gamma_0(s_2)$. Using the procedure described in the previous case, a planning graph can be constructed to find a path that connects p' to q'. Concatenating the resulting curve with $\gamma_0(t)$ for $t \in [0, s_1)$ in the beginning and $\gamma_0(t)$ for $t \in [s_2, 1]$ at the end we get the resulting path. One such construction is shown in Figure 35.

The above discussion makes it clear that it possible to find a path that transports the k-th node at time t_0 from any initial position $x(t_0)$ to a desired position inside the same connected component of $\mathrm{CAD}(X(\mathcal{V}^0))$ in a finite time. In order to give a meaning to this transportation from the point of view of control, it enough to assume that the dynamics of the nodes described by Equation 5 are globally controllable. Combining these observations, we get our main result.

Theorem 10.2.1 *Assume that the individual nodes are globally controllable. Let $\mathcal{V}_i^m = \{k\}$ be fixed. Given two connectivity graphs $\Phi_N(\mathbf{x}(0))$ and G_f, there exists a finite connectivity graph process $\Phi_N(\mathbf{x}(0)) \to G_1 \to \ldots \to G_f$, and a corresponding trajectory $\mathbf{x}(t) \subseteq C^N(\mathbb{R}^2), t \in [t_0, t_f]$ such that $\mathbf{x}(t_f) \in \Phi_N^{-1}(G_f)$ if and only if there exist a finite collection of Nash cells $\mathcal{C}_\mathbf{x} = \{C_0, C_1, \ldots, C_M\} \subseteq \mathrm{CAD}(X(\mathcal{V}^0))$ such that $\mathbf{x}(0) \in C_0$, $\mathbf{x}(t_f) \in C_M$ and $\mathbf{x}(t) \in C_j$ for some $C_j \in \mathcal{C}_\mathbf{x}$ for all $t \in [t_0, t_f]$*

This gives us a way to realize trajectories on the space of connectivity graphs and a method to characterize the reachable set with one mobile node. In fact, by construction this node only needs two different typed of motions, namely along constant vector fields and rotations about fixed points.

10.3 Global Objectives, Desirable Transitions and Optimality

The whole purpose of a coordinated control strategy in a multi-agent system is to evolve towards the fulfilment of a global objective. This typically requires the minimization (or maximization) of a cost associated with each global configuration. Viewed in this way, a planning strategy should basically be a search process over the configuration space, evolving towards this optimum. If the global objective is fundamentally a function of the graphical abstraction of the formation, then it is better to perform this search over the space of graphs instead of the full configuration space of the system. By introducing various graphical abstractions in the context of connectivity graphs, we have the right machinery to perform this kind of planning. In other words, we will associate a cost or score with each connectivity graph and then work towards minimizing it.

Given $\text{Reach}(G_0, \mathcal{V}^m)$, a decision need to be taken regarding what $G_f \in \text{Reach}(G_0, \mathcal{V}^m)$ the system should switch to. For this we define a cost function $\Psi : G_{N,\delta} \to \mathbb{R}$ and we choose the transition through

$$G_f = \underset{G \in \text{Reach}(G_0, \mathcal{V}^m)}{\arg\min} \Psi(G)$$

Here Ψ is analogous to a terminal cost in optimal control. If, in addition, we also take into account the cost associated with every transition in the graph process $G_0 \to G_1 \to \ldots \to G_M = G_f$ that takes us to G_f, then we would instead consider the minimization of the cost

$$J = \Psi(G_f) + \sum_{i=0}^{M-1} \beta(i) L(G_i, G_{i+1}),$$

where $L : \mathcal{G}_{N,\delta} \times \mathcal{G}_{N,\delta} \to \mathbb{R}$ is the analogue of a discrete Lagrangian, $\beta(i)$ are weighting constants, and $G_{i+1} \in \text{Reach}(G_i, \mathcal{V}^m)$ at each step i. The Lagrangian lets us control the transient behavior of the system during the evolution of the graph process. As an example,

let $G_i = (\mathcal{V}_i, \mathcal{E}_i)$ and define

$$L(G_i, G_{i+1}) = |(\mathcal{E}_{i+1} \setminus \mathcal{E}_i) \cup (\mathcal{E}_i \setminus \mathcal{E}_{i+1})|,$$

where $|.|$ gives the cardinality of a set. This Lagrangian is the symmetric difference of the sets of edges in the two graphs. Here, we penalize the addition or deletion of edges at each transition. The resulting connectivity graph process takes G_0 to G_f with minimal structural changes. As another example, if we let $L(G_i, G_{i+1}) = |\mathcal{E}_i| - |\mathcal{E}_{i+1}|$, then the desired G_f is generated while maximizing connectivity during the graph process.

10.4 Open Problem IV: Reachability Analysis with Multiple Dynamic Nodes

The theory developed above should be generalized to multiple interacting dynamic nodes. From a preliminary analysis, we have found that this problem has a strong connection with pursuit-evasion games in a discrete setting. The connection with pursuit-evasion games can be appreciated by recognizing the need to avoid sharing of Nash cells by multiple nodes. We are working towards extending the number of mobile nodes by a decomposition of CAD into non-overlapping regions for each mobile node. It should also be noted that the optimal trajectories obtained in this work are only locally minimizing. A scheme for global optimality is also desirable, but one should keep in mind the decentralized aspects of such a scheme.

PART III

Applications

CHAPTER XI

COVERAGE PROBLEMS IN SENSOR NETWORKS

In this chapter, we give an application of the topological characterization of connectivity graphs developed in Part I. Coverage problems in sensor networks are extremely important in a wide variety of contexts, from cell-phone communications to beacon navigation to security and defense. We restrict attention to systems with stationary nodes in $\mathbb{R}^d$ having radially symmetric coverage domains. Depending on the application, these coverage balls may denote broadcast domains, sensing regions, or visibility domains. Knowing the topology of the union of the coverage domains, in particular the location and morphology of holes, is of significant relevance to the coverage problem.

There are two prominent approaches to coverage problems. The first is perhaps best described as the 'computational geometry' approach, in which one uses the coordinates of the nodes and standard geometric tools (such as the Delaunay or Voronoi diagrams) to determine coverage. For examples of such an approach, see [104, 100, 69, 64]. One feature of this approach is that the precise geometry of the domain and the exact locations of the nodes must be known.

To circumvent some of these difficulties, many researchers have turned to probabilistic methods for coverage. For example, in [60], the author assumes a randomly and uniformly distributed collection of nodes in a domain with a fixed geometry and proves expected area coverage. Other approaches [66, 103, 51] give percolation-type results about coverage and network integrity for randomly distributed nodes. The drawback of these methods is the need for strong assumptions about the exact shape of the domain, as well as the need for a uniform distribution of nodes.

In this chapter we develop a different approach based on the tools of homology theory. Given a collection of nodes $\mathbf{x} \in \mathcal{C}^N(\mathbb{R}^2)$ covered by balls of a fixed radius, we are interesting in knowing the topology of the union of the covering discs. Specifically, are there any holes

in the cover? If so, where are the holes located?

11.1 Problem Formulation

We assume as little as possible about the nodes and their geometry. Consider a collection of stationary nodes $\mathbf{x} \in \mathcal{C}^N(\mathbb{R}^2)$. We adopt the following assumptions on our system.

A1 Nodes have radially symmetric sensing domains of radius r_c.

A2 Nodes broadcast their unique ID numbers. Each robot can detect the identity of anyone within radius r_s via a *strong* signal, and via a *weak* signal within a larger radius r_w.

A3 The radii of communication r_s, r_w and the coverage radius r_c satisfy

$$2r_c = r_w \geq r_s\sqrt{2}. \tag{29}$$

It is important to note what we do not assume. The coordinates of the nodes are unknown to the individual nodes. Neither are there localization or orientation capabilities. Nodes are completely devoid of any information apart from the identities of 'very close' and 'somewhat close' neighbors. This ability to differentiate between strong and weak signals provides a coarse form of distance measurement. For example, if a particular node scans its communications, it can determine whether a certain node is within distance r_s or r_w or neither. The

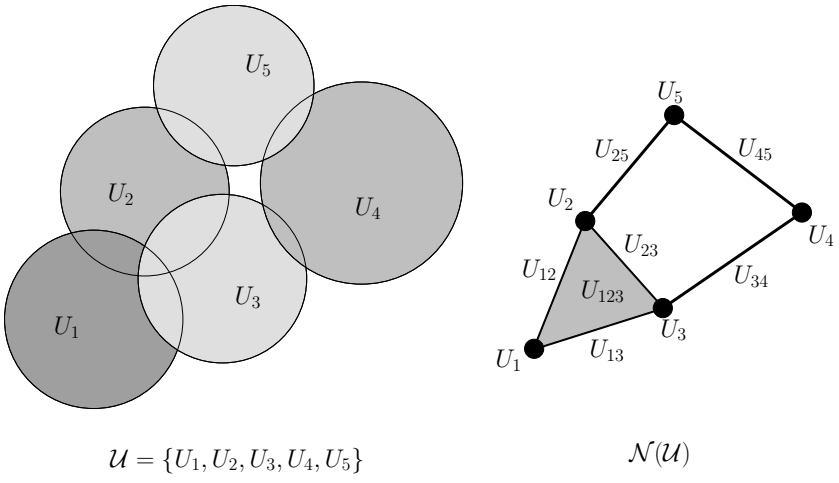


Figure 36: The nerve complex of a collection of sets.

problem of computing the topological type of a union of sets is classical, and easily handled using the concept of a nerve of a collection of sets. We give the following definition.

Definition 11.1.1 *Given a collection of sets $\mathcal{U} = \{U_\alpha\}$, the* nerve complex *of $\mathcal{U}$, $\mathcal{N}(\mathcal{U})$, is the abstract simplicial complex whose k-simplices correspond to nonempty intersections of $k+1$ distinct elements of $\mathcal{U}$.*

Hence, the vertices of $\mathcal{N}$ correspond to the elements of $\mathcal{U}$ themselves. An edge in $\mathcal{N}$ exists between two vertices if and only if the corresponding elements of $\mathcal{U}$ intersect. Higher dimensional simplices are regulated by mutual intersections of collections of elements of $\mathcal{U}$. One such example has been drawn in Figure 36. The nerve of a collection of sets is a an important object in algebraic topology. We state the following classical result.

Theorem 11.1.1 (The Čech Theorem) *The nerve complex of a collection of convex sets has the homotopy type of the union of the sets.*

For a proof, see, e.g., [21]. Unfortunately, nerves are difficult to compute without precise locations of the nodes and a global coordinate system. We instead turn to a different method for obtaining a simplicial complex from a sensor network, using only pairwise communication data. The simplicial complex of choice is the Rips-Vietoris complex, as studied in Chapter 7.

Recall that Rips complexes are completely determined by the connectivity graph of the system: any time you see a triangle in the communication graph, you fill in an abstract 2-simplex. Any time you see a complete subgraph on $k+1$ vertices, you fill it in with an abstract k-simplex.

It would seem reasonable to conjecture that the radius R Rips complex of a set of nodes would be topologically equivalent to the nerve complex of the balls of a particular radius at the nodes. Unfortunately, this is not true. Figure 37 gives an example for which the Rips complex fails to capture the nerve: the nerve is topologically a circle with a single hole in the middle, whereas the Rips complex is an octahedron, which is simply connected. These two complexes have different homology groups and are thus not of the same homotopy type. Similar examples can be constructed for arbitrary choice of coverage radius.

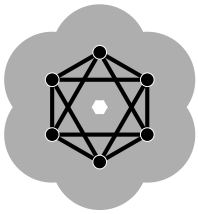


Figure 37: A Rips complex not homotopy equivalent to the union of cover discs.

There is a way to modify the Rips complex to resolve these issues. We define the ℓ^∞ connectivity graph to be the graph whose vertices are the nodes themselves, and whose edges are those pairs of nodes whose ℓ^∞ distance is less than or equal to r_s. Denote by $\mathcal{R}_s^\infty$ the Rips complex of this graph. The following result is very simple.

Lemma 11.1.1 *$\mathcal{R}_s^\infty$ is precisely the nerve complex of the cover of nodes in $\mathbb{R}^d$ by closed d-dimensional cubes of side-length r_s.*

Proof: In the case where $d = 1$, the result is immediate. For the general case, since distances are measured in the ℓ^∞ norm (each coordinate is measured) and cover elements are cubes, the entire problem decomposes into cross-products of the $d = 1$ case. ∎

Unfortunately, it is physically unrealistic to assume that communication can be carried out with precise ℓ^∞ geometry. We therefore consider what can be done with radially symmetric communication and sensing. The core idea is that we can 'squeeze' the ℓ^∞ Rips complex (hence a nerve complex) between two standard Rips complexes of different radii.

11.2 Coordinate-free Detection of Holes

The following theorem is a criterion for hole detection in networks.

Theorem 11.2.1 *A sensor network satisfying assumptions* A1-A3 *has a coverage hole if there is a nonzero homology class in $\mathcal{R}_s$ which is also nonzero as a homology class in $\mathcal{R}_w$. That is, the homomorphism*

$$\iota_* : H_k(\mathcal{R}_s) \to H_k(\mathcal{R}_w) \tag{30}$$

induced by the inclusion $\iota : \mathcal{R}_s \hookrightarrow \mathcal{R}_w$ is nonzero for some $k > 0$.

Proof: We claim that there exists a chain of inclusions

$$\mathcal{R}_s \subset \mathcal{N}_c \subset \mathcal{R}_w, \tag{31}$$

where $\mathcal{N}_c$ is the nerve complex of the cover of $\mathcal{X}$ by balls of radius r_c. Assume for the moment that these inclusions hold, and denote them as $j : \mathcal{R}_s \hookrightarrow \mathcal{N}_c$ and $j' : \mathcal{N}_c \hookrightarrow \mathcal{R}_w$. Since $\iota = j' \circ j$, we conclude that the induced homomorphisms on homology satisfy $\iota_* = j'_* \circ j_*$. Hence, if ι_* is nonzero, then it takes a nonzero element of $H_k(\mathcal{R}_s)$ to a nonzero element of $H_k(\mathcal{R}_w)$, which itself is in the image of j'_*. This implies that $H_k(\mathcal{N}_c) \neq 0$. From the Čech Theorem, we conclude that the cover has a hole of dimension k.

We now demonstrate that Equation (31) holds. Clearly, the inclusion $j' : \mathcal{N}_c \hookrightarrow \mathcal{R}_w$ holds because (since $r_c = r_w/2$) both complexes have the same 1-dimensional skeleton and the Rips complex has all possible simplices of this 1-skeleton filled in.

The inclusion $j : \mathcal{R}_s \hookrightarrow \mathcal{N}_c$ is not as direct. Clearly, $\mathcal{R}_s \subset \mathcal{R}_s^\infty$. From Lemma 11.1.1, $\mathcal{R}_s^\infty$ equals the nerve of the cover of $\mathcal{X}$ by cubes of side length r_s. These cubes are contained inside of balls of radius $r_c \geq r_s\sqrt{d}$. (See Figure 38 for an illustration of this nesting.) Hence, $\mathcal{R}_s \subset \mathcal{N}_c$. Equation (31) holds and the theorem is proved. ∎

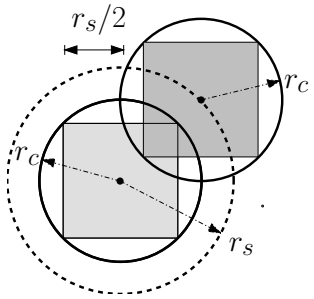


Figure 38: The nesting of sensing and covering discs with the ℓ^∞ balls used in $\mathcal{R}_s^\infty$.

It should be noted that this criterion is a sufficient criterion, but not necessary. In an extreme case, one can choose r_s to be exceedingly small, in which case $H_k(\mathcal{R}_s) = 0$ for all $k > 0$ and the criterion automatically fails.

The proof presented here involves the ℓ^∞-Rips complex. This is not the optimal way to obtain Equation (31). A result of de Silva (see [25]) states that the optimal ratio r_w/r_s for

which the equation holds is $\sqrt{2d/(d+1)}$ for coverage in dimensions $d = 1, 2, 3$. Hence, the results of this note hold with replacing $\sqrt{2}$ in *A3* with the aforementioned constant.

11.3 Simulations

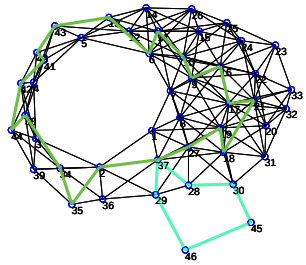
The computation of homology groups for large networks is a nontrivial exercise. The number of simplices in a complex increases exponentially with the dimension of that complex. To make matters worse, the standard algorithm for computing generators in homology is of quintic order in the number of simplices. This may discourage computation of homology in all but very low dimensions. Fortunately, the past few years have witnessed an explosion in algorithms and software for computing homology of simplicial complexes which make nontrivial computations possible (see [55] and references therein).

In order to verify the theoretical results developed in the previous section, we have successfully run several simulations using a publicly available computational homology software known as `CHomP` [23]. These simulations have been written using `MATLAB` as the frontend (primarily for generating the simplicial complexes from various point-data sets, data formatting and for visualization.) The `CHomP` routines have been used for simplification of the Rips complexes, computing the homology groups and for localizing the non-trivial generators of the homology groups.

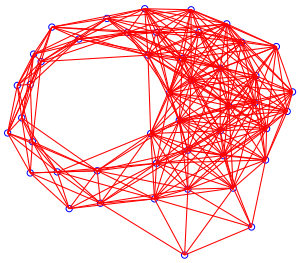
Note that in the figures and examples which follow, we illustrate the cover using coordinates. The frontend keeps track of coordinates for purposes of drawing pictures. However, `CHomP` has no information about coordinates: the homology criterion uses *only* connectivity data as per our assumptions.

Results of one such simulation appear in Figure 39. Figure 39.(a) illustrates the union of sensor coverage discs in $\mathbb{R}^2$, each of radius $r_c = \frac{1}{\sqrt{2}}$. Clearly, there is a hole inside the union of sensor domains. Figure 39.(b) shows the Rips complex $\mathcal{R}_s$ generated by the detection of a strong signal within radius $r_s = 1$. Using `CHomP`, we detect two generators for the first homology group $H_1(\mathcal{R}_s)$ of this complex. Representative cycles for each equivalence class of generators are depicted in the same figure in *green* and *blue*. The generator depicted in *blue* envelops a non-collapsible cycle of 5 nodes in the lower part of the complex whereas

(a) The union of coverage discs in $\mathbb{R}^2$.



(b) Rips Complex $\mathcal{R}_s$.

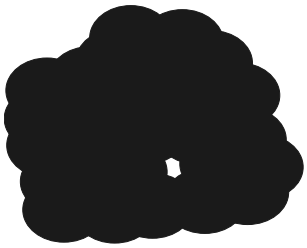


(c) Rips Complex $\mathcal{R}_w$.

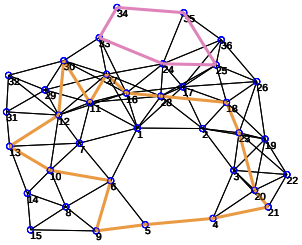
Figure 39: Simulation results with $r_c = \frac{1}{\sqrt{2}}, r_s 1$ and $r_w = \sqrt{2}$.

the generator colored as *green* envelops another non-collapsible cycle of nodes in the left part of the complex. The Rips complex $\mathcal{R}_w$ generated by the detection of a weak signal within $r_w = 2r_c = \sqrt{2}$ appears in Figure 39.(c). First, note that $\mathcal{R}_s \subset \mathcal{R}_w$. A greater radius of detection adds several new edges to the complex, which in turn induce many higher dimensional simplices. The result of the induced homomorphism of the inclusion map $\iota_* : H_1(\mathcal{R}_s) \to H_1(\mathcal{R}_w)$ can be understood as follows. The cycle of nodes enveloped by the *blue* generator in $\mathcal{R}_s$ no longer remains non-collapsible in $\mathcal{R}_w$ due to the addition of many 2-simplices. Therefore the same *blue* generator when seen in $\mathcal{R}_w$ becomes trivial and vanishes. However, the *green* generator remains non-collapsible when seen in $\mathcal{R}_w$, despite the shortening of the non-collapsible cycle it envelops. Therefore, in the language of Theorem 11.2.1, $\iota_* : H_1(\mathcal{R}_s) \to H_1(\mathcal{R}_w)$ is non-zero and indicates the presence of a 1-dimensional hole, as clearly verified from Figure 39.(a).

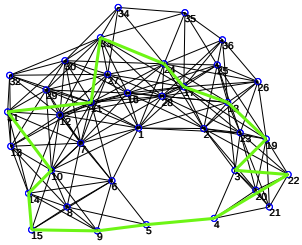
Another such simulation appears in Figure 40. Again, there are two generators, colored as *purple* and *orange*, in $H_1(\mathcal{R}_s)$, each bounding a non-trivial cycle in $\mathcal{R}_s$. However, only one of them survives the inclusion map to $H_1(\mathcal{R}_w)$, showing the presence of a hole. The inclusion map picks the right generator so that the true hole is identified and the false cycle



(a) The union of sensor discs in $\mathbb{R}^2$.



(b) Rips Complex $\mathcal{R}_s$.



(c) Rips Complex $\mathcal{R}_w$.

Figure 40: Simulation results for a second data set with $r_c \frac{1}{\sqrt{2}}, r_s = 1$ and $r_w = \sqrt{2}$.

gets killed.

The homology computation also indicates the location of holes in the network. As pointed out earlier, a cycle representing a generator of a homology group actually represent its entire equivalence class. Two cycles are equivalent if their difference is a boundary. Therefore one can use this property to collapse the generators till they lock exactly onto the non-collapsible cycles they envelop. We denote such cycles as *minimal*. For simplicial complexes, this kind of collapsing or minimization has a nice geometrical interpretation. If two edges of a cycle in dimension 1 belong to a 2-simplex, then one can collapse the cycle to the third edge. This procedure can be generalized to higher dimensions as well [55]. For the purpose of demonstration, such minimization is depicted in Figure 41 for a certain Rips complex generated in our simulations. The non-trivial generator is minimized till it captures the nodes that surround the hole. Although, this method can be applied to all simplicial complexes, it has one serious drawback. For more than one hole in the network, this kind of minimization depends on the initial choice of representative cycles. Ideally, one may wish for each generator surrounding exactly one hole. However, the homology group can possibly be generated equivalently if one hole is enveloped by more than one generators.

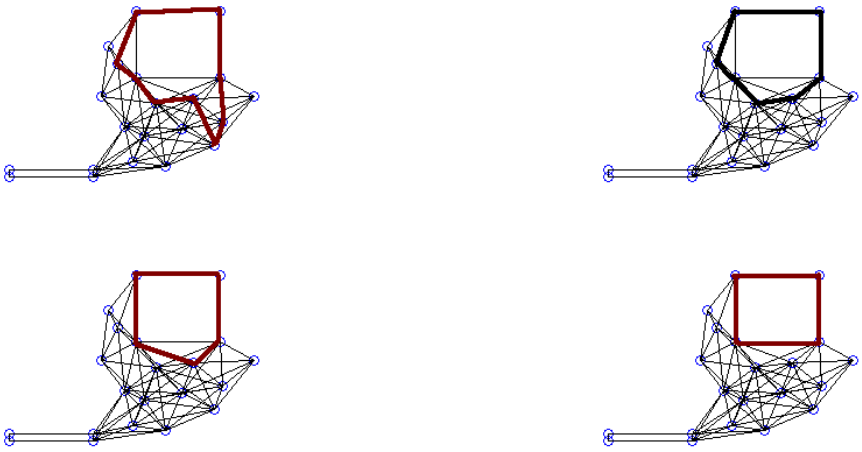


Figure 41: Minimization of generators to detect location of network holes.

Table 1: Run times for five simulations: all times listed in seconds. Here, T^{exp} is the time to build and export the Rips complex (weak and strong respectively) and T^{hom} is the time to compute the homology of the complex.

# Nodes	# Edges	T_w^{exp}	T_s^{exp}	T_w^{hom}	T_s^{hom}
20	45	0.14	0.13	0.49	0.47
67	345	1.46	1.41	0.60	0.60
84	285	3.15	3.11	0.73	0.58
154	1059	26.2	26.1	1.25	0.63
214	1243	92.2	90.2	1.08	0.69

In that case, the minimization would at best partly detect the location of the holes in the network.

All computations were performed on a Pentium 4, 1.60 GHz, 512 Mb RAM, Windows 2000 machine, running Matlab 6.1 as the front end for `CHomP`. Table 1 gives the runtimes for a variety of systems. It is noted that the runtime is dominated by the construction of the Rips complexes: these can have a large number of simplices as a function of the number of nodes. It is a pleasant observation that the homology computation is relatively quick, even though we have not optimized the homology computation code for our (rather particular) type of systems. The number of communications required between nodes is a function of the number of edges in the Rips complex, listed in the second column.

11.4 Generalizations

The methods presented are novel and of potentially great use in sensor networks. The use of topological methods allows one to dispense with assumptions about coordinates, distances, and orientations. The results presented in this note are by no means the best possible results, the reliance on having r_c equal to exactly $r_w/2$ is something that would not necessarily hold in a physical setting. We have presented in this chapter, a limited result for ease of exposition. In another recent work [25], we have developed a homological coverage criterion which is dual to the hole-detection criterion presented here. We refer the interested reader to a series of papers by de Silva et al. [25, 26, 27] that further elaborate on these concepts. We briefly outline the ingredients of this more general theory.

In addition to Assumptions *A1-A3*, there are further assumptions about the domain $\mathcal{D} \subset \mathbb{R}^2$ in which the nodes lie. First, $\mathcal{D}$ is assumed to be bounded with boundary $\partial\mathcal{D}$ which is not too 'pinched'. Second, nodes can detect if they are within some radius r_f of $\partial\mathcal{D}$. These 'boundary' nodes generate sub-complexes $\mathcal{F}_s \subset \mathcal{R}_s$ and $\mathcal{F}_w \subset \mathcal{R}_w$. The result in [25] states that $\mathcal{U}$ contains all of $\mathcal{D}$ (except for possibly a neighborhood of $\partial\mathcal{D}$) whenever the induced map

$$\iota_* : H_2(\mathcal{R}_s, \mathcal{F}_s) \to H_2(\mathcal{R}_w, \mathcal{F}_w)$$

is nonzero. Roughly, $H_k(\mathcal{R}_s, \mathcal{F}_s)$ denotes the k-th relative homology group of the quotient space $\mathcal{R}_s/\mathcal{F}_s$, i.e. the abstract simplicial complexes obtained by collapsing $\mathcal{F}_s$ or $\mathcal{F}_w$ to a single vertex. Similarly, $H_k(\mathcal{R}_w, \mathcal{F}_w)$ denotes the k-th homology group of the quotient space $\mathcal{R}_w/\mathcal{F}_w$. (There is a subtle difference, between $H_k(X, Y)$ and $H_k(X/Y)$. We give a short introduction to the relative homology groups in the Appendix.)

This coverage criterion has several nice features. There are no assumptions on the geometry or topology of the domain, $\mathcal{D}$; for example, it is not assumed that $\partial\mathcal{D}$ is connected. In addition, the radius r_c needs to satisfy only an inequality, not an equality as in this note (since, of course, increasing r_c can only improve coverage).

In order to compute homology relative to the fence sub-complexes, we use the following procedure. To compute $H_2(\mathcal{R}_s, \mathcal{F}_s)$, we begin by adding an abstract vertex to $\mathcal{R}_s$ and then

adding this vertex to every simplex in $\mathcal{F}_s$. This places a cone over $\mathcal{F}_s$, and it yields a complex $\mathcal{Q}(\mathcal{R}_s, \mathcal{F}_s)$ whose homotopy type is that of the quotient space $\mathcal{R}_s/\mathcal{F}_s$. It follows from the Excision Theorem [49] and homotopy invariance that $H_*(\mathcal{R}_s, \mathcal{F}_s) \simeq H_*(\mathcal{R}_s/\mathcal{F}_s) \simeq H_*(\mathcal{Q}(Rs, Fs))$. An example of such a construction is drawn in Figure 43. The fence is drawn in blue, and the additional vertex is drawn on the top, connected by the fence nodes with blue lines. This results in a cone, by adding in higher simplices, where appropriate. In this example, the quotient space becomes homotopic to a punctured sphere (mainly due to the network hole in the original complex), which in turn is homotopic to a point. Hence $H_k(\mathcal{R}, \mathcal{F}) = 0$ for $k > 0$.

These results have been verified using simulations, that have been written using MATLAB as the frontend, primarily for generating the simplicial complexes from various point-data sets, data formatting and for visualization. The `Plex` package has been used for computing homology. `CHomP` has been used to return explicit generators, which gives more precise information than `Plex`, at the cost of running much more slowly. An examples of successful application of the homological criterion appears in Figure 44. The domains is not simply-connected (in other words, it has multiple boundary components). The 172 nodes are presented as embedded in the domain $\mathcal{D}$ and the cover $\mathcal{U}$ is illustrated. Note that the the cover is not too redundant. There are regions which are covered by only one node. These simulations were run on a Linux/PC 1-Gbyte Memory Dual Processor Intel Xeon CPU 1700MHz; cache size 256 KB; MATLAB ver 6.5. The run time for Plex to compute the existence of a nontrivial persistent homology generator is roughly 16 seconds. The vast majority of the run time is spent constructing the simplicial complexes from the input data: the actual persistence computation is much faster. The complex $\mathcal{R}_w$ contains a total of 135295 3-d simplices. An example of a system for which the homology criterion gives a false negative has been drawn in Figure 45. Note the fragility of the cover in the upper left portion, as one could predict by the quadrilateral 1-cycle in $\mathcal{R}_s$.

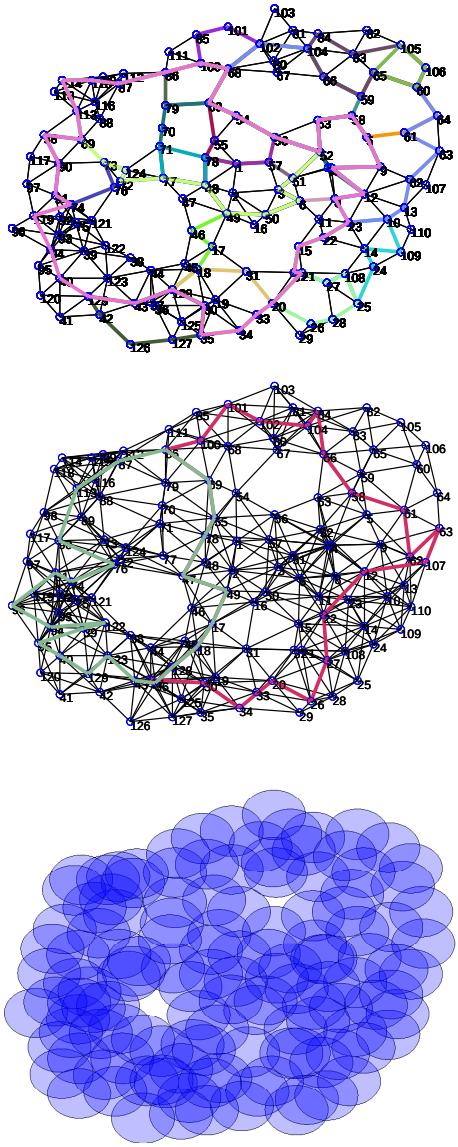


Figure 42: Network Hole Detection in network of 129 nodes: [top] $\mathcal{R}_s$, [center] $\mathcal{R}_w$, [bottom] $\mathcal{U}$.

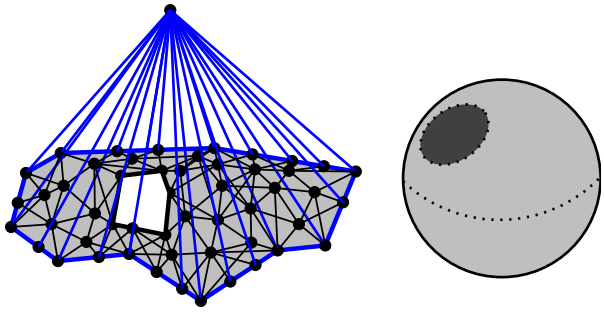


Figure 43: An example of placing a cone over $\mathcal{R}$ for computing $H_*(\mathcal{R}, \mathcal{F})$. The result is a quotient space $\mathcal{R}/\mathcal{F}$, which is homotopic to a punctured sphere.

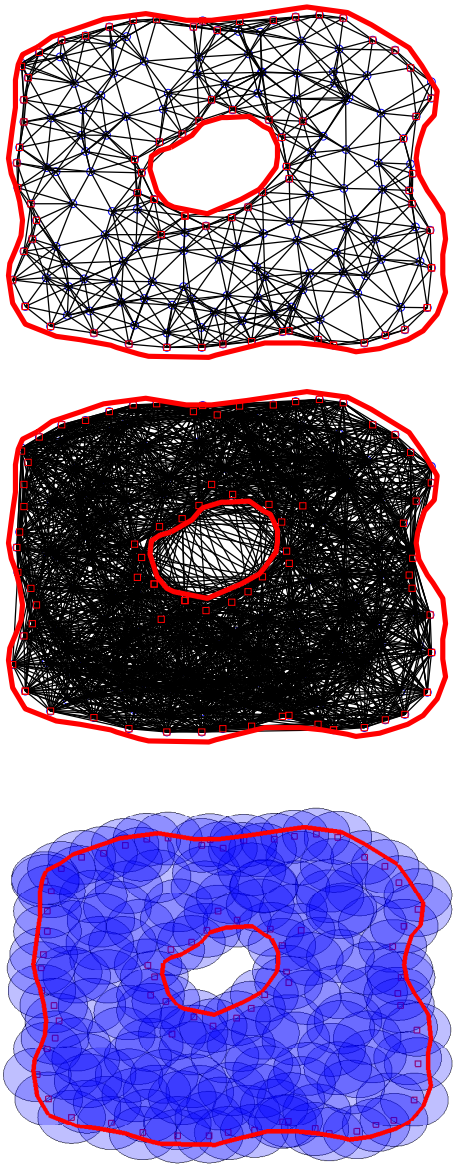


Figure 44: An example of a system of 172 nodes in a domain with multiple boundary components for which the homological criterion is satisfied: [left] $\mathcal{R}_s$, [center] $\mathcal{R}_w$, [right] $\mathcal{U}$.

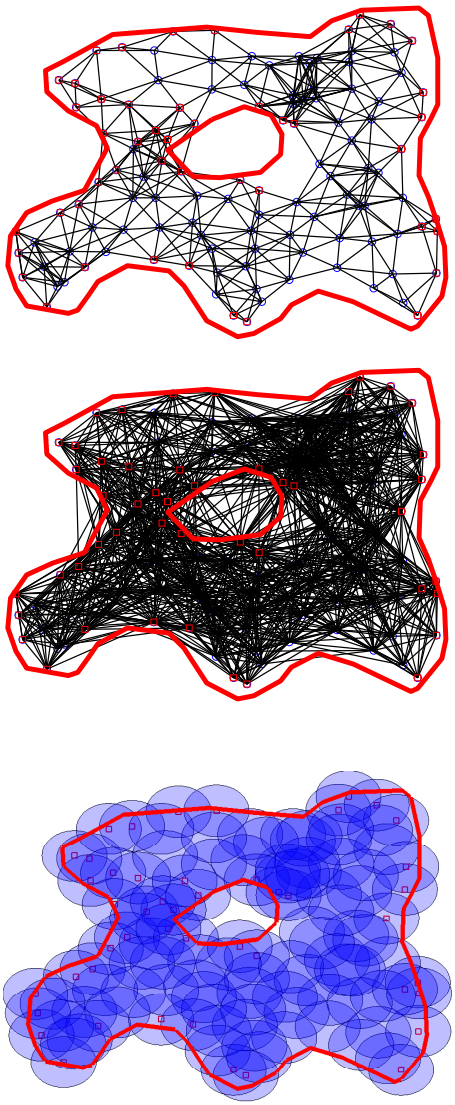


Figure 45: An example of a system for which the homology criterion gives a false negative.

CHAPTER XII

PRODUCTION OF LOW-COMPLEXITY FORMATIONS

12.1 δ-Chains with No Movements

The geometric structure and topological characterization of the connectivity graphs of multi-agent formations given in Part I, can be used to obtain certain global properties of formations using decentralized algorithms, suitable for implementation on scalable, totally decentralized multi-agent systems. It will be appropriate to mention, that the entire machinery presented in Chapters 6, 7 and 8 for converting connectivity graphs into simplicial complexes becomes irrelevant, if the computations are performed in a centralized manner. The centralized method of obtaining the simplicial subgraph can be much simpler and need not have knowledge of the local geometrical structure obtained above. We therefore have the following corollary of the main theorem in Chapter 8.

Corollary 12.1.1 *The maximal simplicial subgraph G^* of a connectivity graph $G \in \mathcal{G}_{N,\delta}$ can be obtained using a decentralized algorithm.*

The detection of genus is also implementable as a decentralized algorithm. Since the fundamental group $\pi_1(I_{M_{G^*}})$ is isomorphic to the edge group $E(M_{G^*})$ of the triangulation (see the Appendix for details), we can base our method on the reduction of loops inside M_{G^*}, according to the equivalence rules of $E(M_{G^*})$ [5]. Using this approach, we can implement a decentralized algorithm to find out if M_{G^*} has genus 0 or not. The role of low-complexity formations called δ-chains has been studied and emphasized in Chapter 5. Decentralized algorithms for obtaining these δ-chains are of considerable interest. It can be shown there that the ability to obtain a simplicial representation of graphs, lets us obtain δ-chains for an important class of connectivity graphs, namely the graphs whose maximal simplicial complex has genus 0. Some snapshots of the algorithm have been given in the figures. The maximal simplicial subgraph is obtained during the initial steps of the algorithm, followed

by the determination of genus (i.e. whether it has a genus 0 or not), and finally the evolution of the δ-chain.

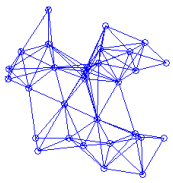


Figure 46: A connectivity graph.

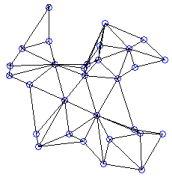


Figure 47: Maximal simplicial subgraph of connectivity graph of 46.

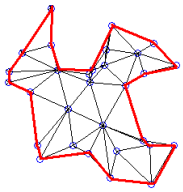


Figure 48: Boundary detection

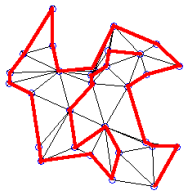


Figure 49: Resulting δ-chain.

12.2 Connectivity Graph Processes that Generate δ-Chains

We now study a particular type of multi-agent coordination problem, i.e. the production of low-complexity formations. In a previous chapter, we have presented a complexity measure for studying the structural complexity of robot formations. The structural complexity is based on the number of local interactions in the system due to perception and communication. When designing control strategies for distributed, multi-agent systems, it is vitally important that the number of prescribed local interactions is managed in a scalable manner. In other words, it should be possible to add new robots to the system without causing a significant increase in the communication and computational burdens of the individual robots. On the other hand, an additional requirement when designing multi-agent coordination strategies should be that enough local interactions are present in order to ensure the proper execution of the task at hand. It turns out that the notion of structural complexity is the right measure to compare these conflicting requirements for the system. Therefore, it would be desirable to obtain graph processes that transform a formation with a high structural complexity to one with a lower complexity (and vice versa.) Recall that we have

defined the structural complexity of a connectivity graphs G as

$$C(G) = \sum_{v_i \in \mathcal{V}} \left(\deg(v_i) + \sum_{v_j \in V, v_i \neq v_j} \frac{\deg(v_j)}{d(v_i, v_j)} \right),$$

where $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ is some distance function defined between vertices. It was also observed that if G is a connected connectivity graph then the complexity of G is bounded above and below by

$$C(\delta_N) \leq C(G) \leq C(\mathcal{K}_N),$$

where δ_N is the δ-chain on N vertices, and $\mathcal{K}_N$ is the complete graph. The lowest complexity graphs or δ-chains (which are the line graphs or the Hamiltonian paths on all vertices), are important for formations that require minimal coordination. Therefore coming up with a planning mechanism to produce low-complexity formations from an arbitrary initial formation is a useful result in multi-agent coordination. Using the concepts from previous sections we define $\Psi(G) = C(G)$ and $L(G_i, G_{i+1}) = |(\mathcal{E}_{i+1} \setminus \mathcal{E}_i) \cup (\mathcal{E}_i \setminus \mathcal{E}_{i+1})|$, where $G_i = (\mathcal{V}_i, \mathcal{E}_i)$. In this way we tend to produce formations that are lower in complexity by generating a graph process that adds or deletes a small number of edges at each transition. The results of one such simulation has been shown in Figure 50. Here, the star-like graph in the upper left corner is the initial connectivity graph (with a higher structural complexity than a δ-chain on 5 vertices). The graph process evolves from left to right and then continues onto the lower rows in the same manner until it reaches the δ-chain in the lower right corner. The process involves transitions to various intermediate graphs. In this example, the mobile node is labelled as 1. As predicted by the CAD decomposition, it first slides up to make an edge with node 2 and then rotates about node 2. It then passes by node 5 making various intermediate graphs, till it comes in the vicinity of node 4. Finally it makes a rotation about node 5 to form the δ-chain. Also note that due to the choice of the Lagrangian described above, the mobile node makes (or breaks) only a minimum number of edges at each graph transition. In this example, it can be seen that this number is always 1.

A more elaborate example is shown in Figure 51, where the δ-chain is achieved by a series of maneuvers in which different nodes take the role of the mobile node at appropriate steps. The resulting connectivity graph process is a concatenation of the graph processes

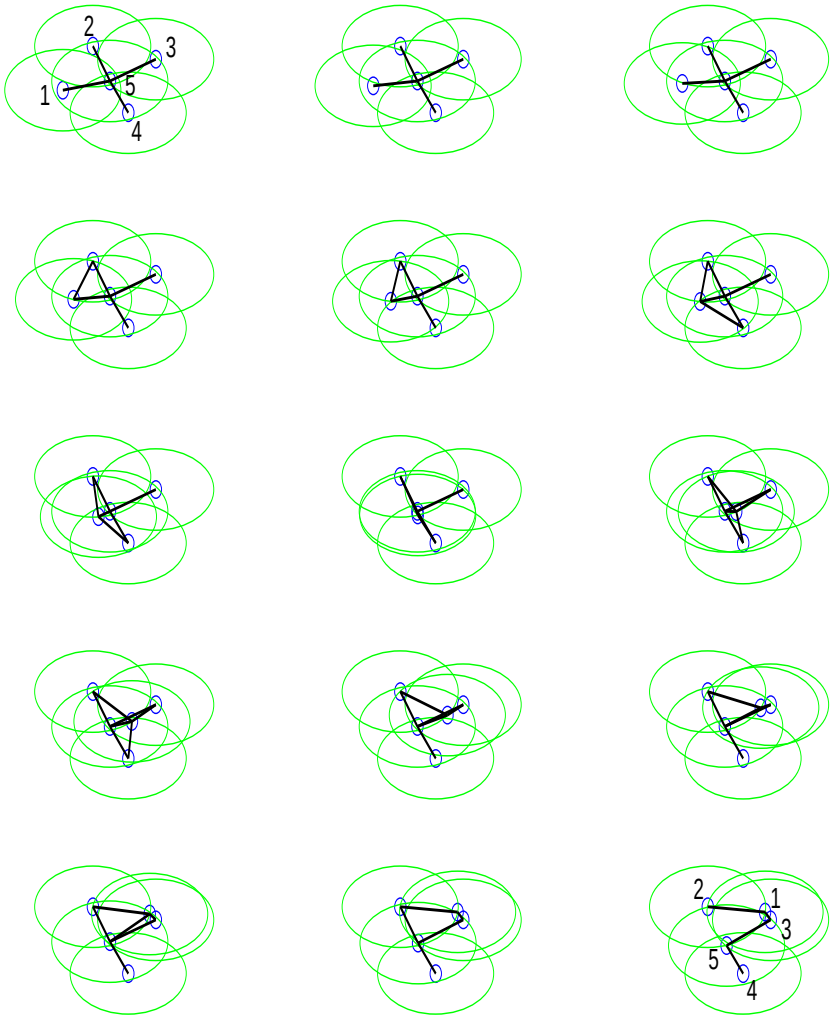


Figure 50: A connectivity graph process that generates a δ-chain.

described in above. Simulations for a relatively large number of nodes and more complex formations have also been done.

12.3 Open Problem IV: Motion Description Languages (MDLs) in Graph Processes

It should be noted that the path planning algorithm developed in Chapter 10 is implementable on a team of robots that have only two modes of motions, namely the linear motion and rotation about a fixed point. One can ask if the reachability sets can be expanded using more complex modes of motion. This ties nicely with a motion description language (MDL) framework [50] in the context of hybrid dynamical systems.

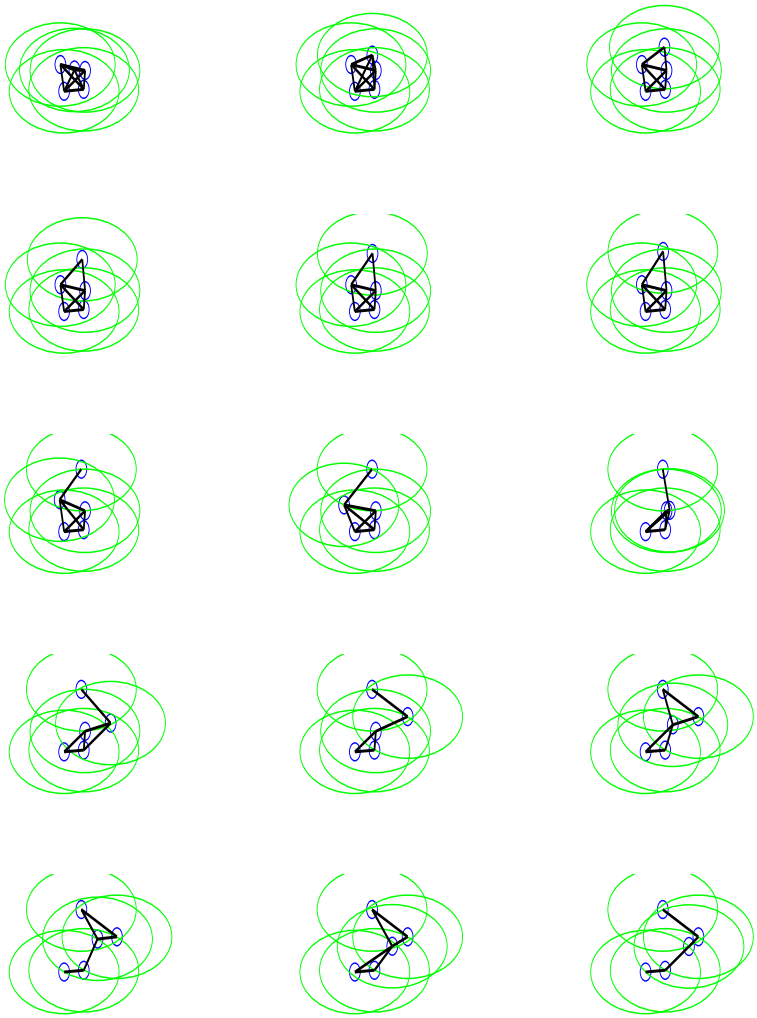


Figure 51: Snapshots of a connectivity graph process that generates a δ-chain by choosing different mobile nodes at appropriate intermediate transitions.

CHAPTER XIII

COLLABORATIVE BEAMFORMING IN SENSOR NETWORKS

Another promising application of the framework presented in Chapter 9 is collaborative beamforming. In ad-hoc and wireless sensor networks, long range communication between clusters is always expensive due to limitations on power and communication channels. Collaborative beamforming is one way to solve this problem without dramatically increasing the complexity compared to non-collaborative strategies [84, 2].

With this method, a cluster of sensors synchronize their phases and collaboratively transmit or receive data in a distributed manner. By properly designing the array factor, one can shape and steer the beam pattern in such a way that the array has either a high power concentration in the desired direction with little leakage (when transmitting), or a high gain in the direction of arrival (DOA) of the signal of interest with significant attenuation in the direction of interference (when receiving). These properties enables Space-Division Multiplex Access (SDMA) among clusters. In most beamforming applications, the array geometry is assumed to be fixed and the optimal beam pattern is formed by optimally weighing the signals received at individual nodes [101]. In this work, we further optimize the beam pattern by altering the geometry of the sensor array using connectivity graph processes.

It should be mentioned that finding an optimal geometry is a difficult design problem in array signal processing. Most designs favor a regular equispaced geometry such as linear, circular, spherical and rectangular grid arrays over random geometries[101]. If one follows the design philosophy in the array processing community, one would tend to drive all nodes to a regular geometry for obtaining better beam patterns. However, more exotic geometries have also been designed for particular applications. Moreover, the placement of nodes in a sensor network is not merely to optimize network communication, but also to maximize

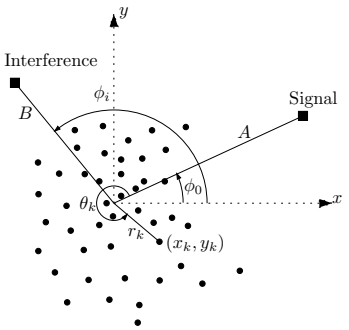


Figure 52: Sensor array geometry.

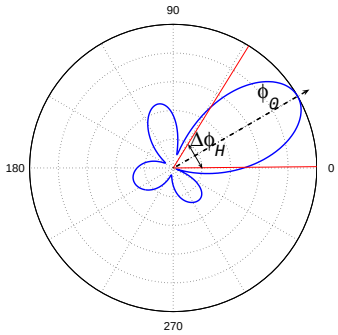


Figure 53: Beam pattern and HPBW.

some benefit associated with distributed sensing. Therefore, it seems beneficial to optimize the geometries over some cumulative function of both the communication performance as well as the sensing performance of the network. We present this approach below.

13.1 Problem Formulation

Let us first study the beamforming for an arbitrary geometry. Following the standard notation in the array signal processing literature, we describe the positions of the individual nodes, signal and interference in polar coordinate system. The signal is located at a distance A and azimuthal angle ϕ_0 while the interference is at an angle ϕ_i as shown in Figure 52. The position $\mathbf{x}_k = (x_k, y_k)$ of the k-th node is given by (r_k, θ_k), where $r_k = \sqrt{x_k^2 + y_k^2}$ and $\theta_k = \tan^{-1}(y_k/x_k)$. Given the position of the sensor array $r = [r_1, r_2, \ldots, r_N]$, $\theta[\theta_1, \theta_2, \ldots, \theta_N]$, we adopt the beamforming algorithm presented in [84], where the gain in direction ϕ is given by the norm of the array factor θ

$$F(\phi|r,\theta)e^{j\frac{2\pi}{\lambda}A}\frac{1}{N}\sum_{k=1}^{N}e^{j\frac{2\pi}{\lambda}r_k[\cos(\phi_s-\theta_k)-\cos((\phi-\theta_k))]}.$$

For known signal and interference directions, the objective of beamforming is to obtain high signal to interference ratio (SIR) and fine resolution, i.e. we would like to keep the main lobe of the beam pattern as thin as possible while minimizing the power in the interference direction. Let $\Delta\phi_H$ be the half power beam width (HPBW) of the main lobe as depicted in Figure 53. The power concentrations (accumulated gain) in the direction of signal and

interference are respectively given by

$$P_s(r,\theta) = \int_{\phi_0-\Delta\phi_H/2}^{\phi_0+\Delta\phi_H/2} |F(\phi|r,\theta)|^2 d\phi, \quad P_i(r,\theta) \int_{\phi_i-\Delta\phi_H/2}^{\phi_i+\Delta\phi_H/2} |F(\phi|r,\theta)|^2 d\phi.$$

We choose the following metric to evaluate the performance of a sensor array geometry.

$$J'(r,\theta) \triangleq \frac{P_s(r,\theta)}{P_i(r,\theta)\Delta\phi_H} = J'(\mathbf{x}),$$

where $\mathbf{x}$ is the sensor array geometry in cartesian coordinates. We wish to maximize the value of this metric over various geometries. To elaborate this point further, we give some example geometries and their respective beam patterns in Figure 55. Here, we assume that the signal direction ϕ_0 is 0 degrees and the interference is coming at an azimuthal angle $\phi_i = 90$ degrees. For comparison, the values of the metric have also been given on top of the beam patterns. Note that the linear array has a narrower beam but a large leakage in the interference direction. Similarly, the circular geometry in the bottom has low interference but a fat beam (i.e a large $\Delta\phi_H$) in the direction of signal. The irregular patterns in the middle have a higher benefit, although they lie in between the two extremes of beam width and interference nullification. Moreover, as described in the above paragraphs, the metric to extremize may not be a function of the beamforming performance alone. Therefore it is reasonable to search over all possible geometries, rather than driving all nodes to a pre-determined regular geometry.

13.2 Improvement in Beamforming Using Graph Processes

Note that the metric defined above may be different for different realizations of a particular connectivity graph. We make use of the cylindrical algebraic decomposition (CAD) algorithm for computing reachability to get a representative geometry r, θ for the connectivity graph G. In this way we let $J'(\mathbf{x}) = J'(G)$, where $\Phi_N(\mathbf{x}) = G$. If $J''(G)$ is some other performance metric associated with the function of the sensor network, then using the notation in the previous chapter, $\Psi(G) = \nu_1 J'(G) + \nu_2 J''(G)$. Similarly we chose $L(G_i, G_{i+1})$ according to a desired transient behavior in the network. We give a snapshot from one such simulation for a particular choice of cost metric in Figure 55. Here, we have purposely chosen a small number of nodes and a relatively less relative displacement to demonstrate

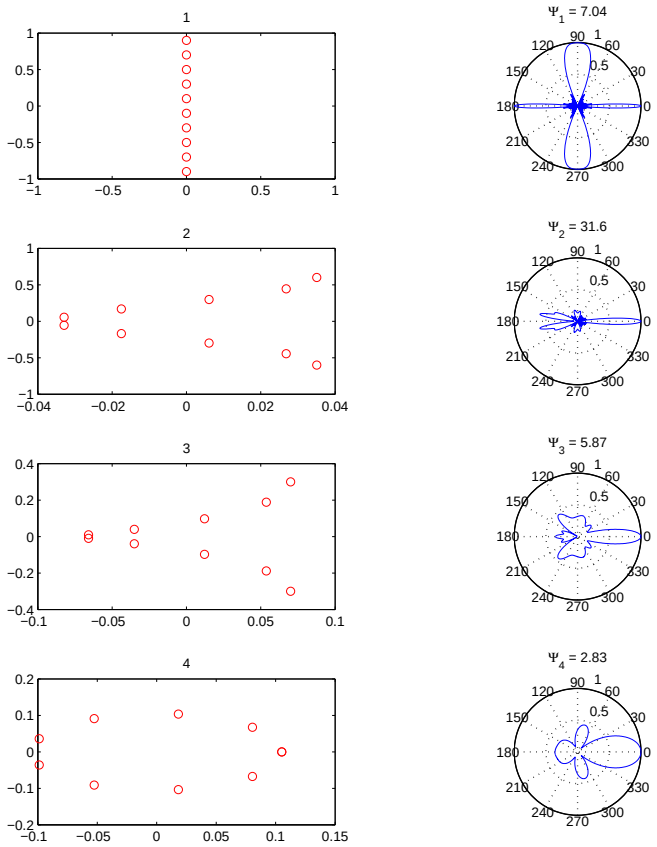


Figure 54: Beamforming performance for various geometries.

the value of the graph processes. The signal and interference directions are the same as in Figure 54. As the graph process evolves, notice the thinning of the main lobe in the signal direction. More pronounced is the attenuation in the interference direction, thus increasing the value of the metric at each transition.

13.3 Further Applications

In principle, the framework developed in Chapter 9 can be used for any application that required optimization over connectivity graphs. We have presented such several such applications in a recent work [80]. It is possible to use the connectivity graph process framework for coverage enhancement in sensor networks, using the homological coverage criterion developed in the previous chapter. This, along with many other applications, is a subject of

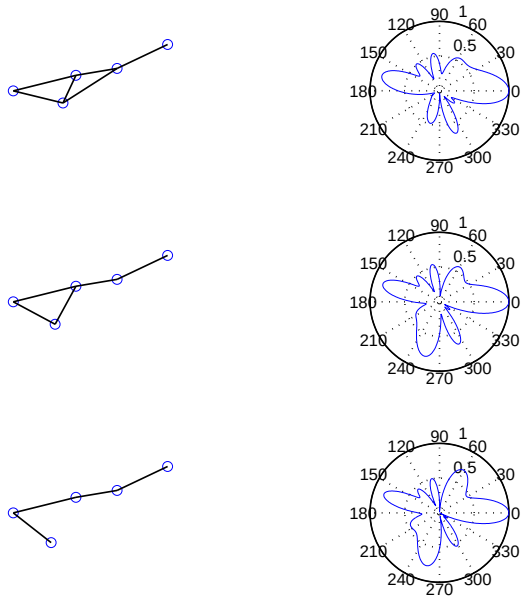


Figure 55: Evolution of beam pattern in a connectivity graph process.

current investigation.

APPENDIX A

NOTES ON HOMOLOGY THEORY

We provide a short introduction to the various concepts of homology theory in algebraic topology. We closely follow the presentation in [49]. Interested readers are referred to this excellent text for a comprehensive introduction. We start with a quick review of some frequently used concepts of elementary group theory.

A.1 Free Abelian Groups

Let $(G_1, +)$ and $(G_2, *)$ be two Abelian groups. A map $f : G_1 \to G_2$ is said to be a *homomorphism* if

$$f(x + y) = f(x) * f(y),$$

for any $x, y \in G_1$. A bijective homomorphism is called an *isomorphism*. We write this as $G_1 \simeq G_2$. The *fundamental theorem of homomorphism* is stated as follows.

Theorem A.1.1 *Let $f : G_1 \to G_2$ be a homomorphism. Then*

$$G_1/ker \simeq imf.$$

Take r elements $g_1, \cdots, g_r$ of a group G. The elements of G of the form

$$n_1 g_1 + \cdots + n_r g_r,\ n_i \in \mathbb{Z},\ 1 \leq i \leq r,$$

make a subgroup H inside G. H is said to be a subgroup generated $g_1, \cdots, g_r$. If G itself is generated by a finite number of elements, then G is said to be finitely generated. The elements $g_1, \cdots, g_r$ are said to be linearly independent if $n_1 g_1 + \cdots + n_r g_r = 0$ only when $n_1 = \cdots = n_r = 0$. If G is finitely generated by r linearly independent elements, G is called a *free Abelian group* of rank r.

If G is generated by one element g, $G = \{0, g, 2g, \cdots\}$ is called a *cyclic group*. If $ng = 0$ for some $n \in \mathbb{Z} - \{0\}$, then G is a *finite cyclic group*. Otherwise, it is an *infinite cyclic*

group. Any infinite cyclic group is isomorphic to $\mathbb{Z}$, while a finite cyclic group is isomorphic to some $\mathbb{Z}_n$. If G is free Abelian group of rank r and H is subgroup of G. We may choose p generators $g_1, \cdot, g_p$ out of r generators of G so that $k_1 g_1, \cdots, k_p g_p$ generate H of rank p. In other words

$$H \simeq k_1\mathbb{Z} \oplus k_2\mathbb{Z} \oplus \cdots \oplus k_p\mathbb{Z}.$$

We now give the *fundamental theorem of finitely generated Abelian groups*.

Theorem A.1.2 *Let G be a finitely generated Abelian group (not necessarily free) with m generators. Then G is isomorphic to the direct sum of cyclic groups,*

$$G \simeq \underbrace{\mathbb{Z} \oplus \mathbb{Z} \oplus \cdots \oplus \mathbb{Z}}_{r} \oplus \mathbb{Z}_{k_1} \oplus \cdots \oplus \mathbb{Z}_{k_p},$$

where $m = r + p$. r is called the rank of G.

Finally an *exact sequence* is defined as a sequence of Abelian groups and homomorphisms between them,

$$\cdots \to A_{n+1} \xrightarrow{\alpha_{n+1}} A_n \xrightarrow{\alpha_n} A_{n-1} \to \cdots$$

such that $\ker \alpha_n = \text{im } \alpha_{n+1}$ for each n.

A.2 Topological spaces and Homotopy

Let X be any set and J be an index set. Let $\mathcal{U} = \{U_i \mid i \in J\}$ denote a certain collection of open subsets of X. The pair $(X, \mathcal{U})$ is a *topological space* of $\mathcal{U}$ satisfies the following:

1. $\emptyset, X \in \mathcal{U}$.
2. If I is a (possibly infinite) sub-collection of J, then $\cup_{i \in I} U_j \in \mathcal{U}$.
3. If K is any finite sub-collection of J, then $\cap_{k \in K} U_k \in \mathcal{U}$.

A *deformation retract* of a topological space X onto a subspace A is a family of maps $f_t : X \to X$, $t \in [0, 1]$, such that $f_0 = \text{id}$ (the identity map), $f_1(X) = A$ and $f_t|_A = \text{id}$ for all t. The family f_t, should be continuous in the sense that the associated map $X \times [0, 1] \to X$, $(x, t) \mapsto f_t(x)$ is continuous.

A deformation retract is a special case of the general notion of homotopy. A *homotopy* is simply any family of maps $f_t : X \to Y$, $t \in [0,1]$, such that the associated map $F : X \times [0,1] \to Y$ given by $F(x,t) = f_t(x)$ is continuous. Two maps $f_0, f_1 : X \to Y$ are said to be *homotopic maps*, if there exists a homotopy f_t connecting them and once writes $f_0 \simeq f_1$. In these terms a deformation retract of X onto a subspace A is a homotopy from the identity map of X onto A, a map $r : X \to X$ such that $r(X) = A$ and $r|_A = \mathrm{id}$.

In this work, we are mainly concerned with a special type of topological spaces, known as simplicial complexes. For an introduction to simplicial complexes, see Chapter 7. Here, we introduce some broader classes of topological spaces, namely the CW complexes and Δ-complexes. Simplicial complexes are special types of these spaces. We therefore present the general theory for a more comprehensive introduction.

A *cell complex* is a topological space constructed by the following procedure:

1. Start with a discrete set X^0, whose points are regarded as 0-cells.

2. Inductively, from the *n-skeleton* X^n, construct X^{n-1} by attaching n-cells e^n_α via maps $\phi_\alpha : S^{n-1} \to X^{n-1}$. This means that X^n is the quotient space of the disjoint union $X^{n-1} \coprod_\alpha D^n_\alpha$ of X^{n-1} with a collection of n-disks D^n_α under the identifications $x \sim \phi_\alpha(x)$ for $x \in \partial D^n_\alpha$. Thus as a set, $X^n = X^{n-1} \coprod_\alpha e^n_\alpha$ where each e^n_α is an open n-disk.

3. Once can either stop this inductive process at a finite stage, setting $X = X^n$ for some $n < \infty$, or one can continue indefinitely, setting $X = \cup_n X^n$. In the latter case X is give the weak topology: A set $A \subset X$ is open (or closed) if and only if $A \cap X^n$ is open (or closed) in X^n for each n.

Cell complexes are also called as CW complexes. An example of a cell complex is drawn in Figure 56. This cell complex has one 0-cell, two 1-cells and one 2-cell. The sphere S^n has the structure of a cell complex with just two cells, e^0 and e^n, the n-cell being attached by the constant map $S^{n-1} \to e^0$.

Each n-cell e^n_α in a cell complex has a *characteristic map* $\Phi_\alpha : D^n_\alpha \to X$ which extends the attaching map ϕ_α and is a homomorphism from the interior of D^n_α onto e^n_α. Therefore,

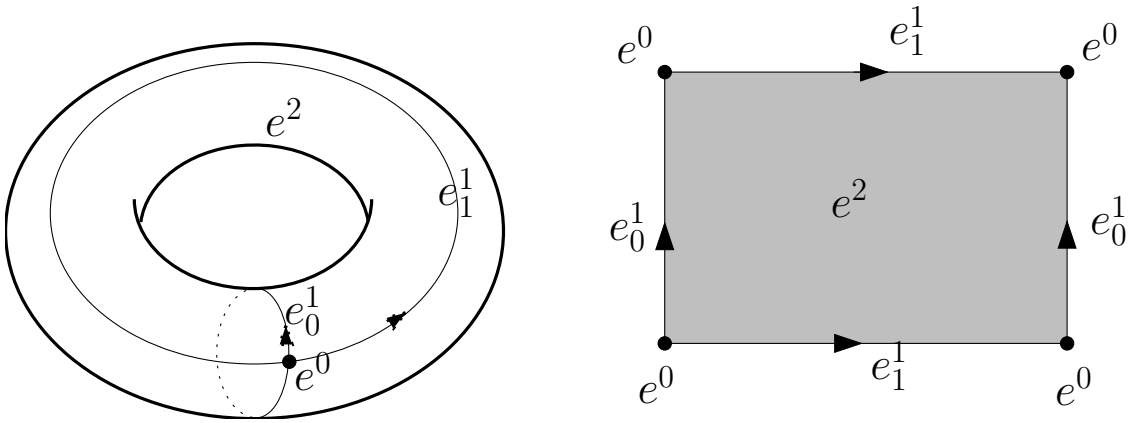


Figure 56: A cell complex representation of a torus $S^1 \times S^1$.

Φ_α can be thought of as the composition

$$D^n_\alpha \hookrightarrow X^{n-1} \coprod_\alpha D^n_\alpha \xrightarrow{\pi} X^n \hookrightarrow X.$$

where π is the quotient map defining X^n.

A *sub-complex* of a cell complex is a closed subspace $A \subset X$ that is a union of cells of X. A is a cell complex in its own right. A pair (X, A) consisting of a cell complex X and a sub-complex A is called a *CW pair*.

A graph is a 1-dimensional cell complex. It contains vertices (0-cells) and edges (1-cells). Similarly, simplicial complexes can also be thought of as cell complexes. It is however, more instructive to start with a more primitive form of complexes, known as Δ-complexes.

Δ-complexes are built out of simplices. An *n-simplex* is defined as the smallest convex set in $\mathbb{R}^d$ containing $n+1$ points $v_0, \cdots, v_n$, that do not lie in a hyperplane of dimension less than n. The points v_i are called the *vertices* of the simplex, and the simplex itself is denoted by $[v_0, v_1, \cdots, v_n]$. The standard n-simplex is given by

$$\Delta^n = \{(t_0, \cdots, t_n) \in \mathbb{R}^{n+1} | \sum_i t_i = 1, \text{ and } t_i \geq 0 \text{ for all } i\}.$$

A *face* of a simplex $[v_0, \cdots, v_n]$ is the sub-simplex with vertices any nonempty subset of the v_i's. By convention, a face is ordered according to their order in the larger simplex. A Δ-*complex* X, is a quotient space of a collection of disjoint simplices obtained by identifying certain of their faces via the canonical linear homeomorphisms that preserve the ordering of vertices. Hence, the identifications never result in two distinct points in the interior of

a face, being identified in X. Therefore, X is the disjoint union of a collection of open simplices (simplices with their proper faces deleted).

Each such open simplex e^n_α of dimension n comes equipped with a canonical map (called the characteristic map) $\sigma_\alpha : \Delta^n \to X$ restricting to a homeomorphism from the interior of Δ^n onto e^n_α. A key property of the characteristic map is that its restrictions to $(n-1)$-dimensional faces of Δ^n are characteristic maps σ_β for open simplices e^{n-1}_β of X. This property can be used to define a Δ-complex as a CW complex X in which each n-cell e^n_α has a distinguished characteristic map $\sigma_\alpha : \Delta^n \to X$ such that the restriction of σ_α to each $(n-1)$-face of Δ^n is the distinguished characteristic map of an $(n-1)$-cell of X.

A.3 Simplicial Homology

We first define the simplicial homology of Δ-complexes. For a more gentle introduction, see Chapter 7. Let $\Delta_n(X)$ ne the free Abelian group with basis the open simplices e^n_α of the Δ-complex X. The elements of $\Delta_n(X)$ are called as n-chains. These elements can be written as finite sums $\sum_\alpha n_\alpha e^n_\alpha$ with coefficients $n_\alpha \in \mathbb{Z}$. One can also consider them as $\sum_\alpha n_\alpha \sigma_\alpha$.

The *boundary homomorphism* $\partial_n : \Delta_n(X) \to \Delta_{n-1}(X)$ can be defined by specifying its values on basis elements:

$$\partial_n(\sigma_\alpha) = \sum_i (-1)^i \sigma_\alpha|_{[v_0,\cdots,\hat{v_i},\cdots,v_n]}.$$

Lemma A.3.1 *The composition* $\Delta_n(X) \xrightarrow{\partial_n} \Delta_n(X) \xrightarrow{\partial_{n-1}} \Delta_{n-2}(X)$ *is zero. In other notation* $\partial_n \circ \partial_{n-1} = 0$.

Proof: This can be checked by a simple calculation.

$$\begin{aligned}
\partial_{n-1}(\partial_n(\sigma)) &= \partial_{n-1}\left(\sum_i (-1)^i \sigma|_{[v_0,\cdots,\hat{v_i},\cdots,v_n]}\right) \\
&= \sum_{j<i} (-1)^i (-1)^j \sigma|_{[v_0,\cdots,\hat{v_j},\cdots,\hat{v_i},\cdots,v_n]} + \sum_{j>i} (-1)^i (-1)^{j-1} \sigma|_{[v_0,\cdots,\hat{v_i},\cdots,\hat{v_j},\cdots,v_n]} \\
&= 0.
\end{aligned}$$

The chain groups $\Delta_n(X)$ are generally denoted by C_n. Note that each of the chain groups

C_n is an Abelian group. We therefore get a sequence of homomorphisms of Abelian groups

$$\cdots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \cdots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

with $\partial_k \partial_{k+1} = 0$ for each k. Such a sequence is called a *chain complex*. From $\partial_k \partial_{k+1} = 0$ it follows that $\text{im}\partial_{n+1} \subset \ker\partial_n$. We define the *simplicial homology groups* by the quotient groups

$$H_n^{\Delta}(X) = \frac{\ker\partial_n}{\text{im}\partial_{n+1}}.$$

The elements of $H_n^{\Delta}(X)$ are the cosets of $\text{im}\partial_{n+1}$, and are referred to as *homology classes*. Elements of $\ker\partial_n$ are called as *cycles* and those of $\text{im}\partial_{n+1}$ are called as *boundaries*. Two cycles representing the same homology class are said to be *homologous*.

A.4 Example Computations of Simplical Homology

The dimension of $H_0^{\Delta}(X)$, is equal to the number of path-connected components of X. The simplest basis for $H_0^{\Delta}(X)$ consists of a choice of vertices in X, one in each path-component of X. Likewise, the simplest basis for $H_1^{\Delta}(X)$ consists of loops in X, each of which surrounds a different 'hole' in X. For example, if X is a graph, then $H_1^{\Delta}(X)$ is a measure of the number and types of cycles in the graph. These concepts can be understood more clearly with the following example.

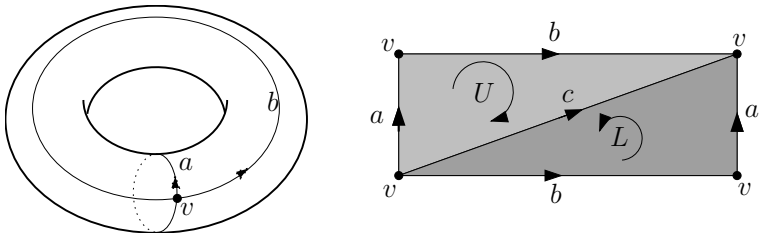


Figure 57: A torus [Left] and a Δ-complex [Right] corresponding to its triangulation T.

In Figure 57, a hollow doughnut-like two-dimensional surface, called a torus has been drawn. Imagine that we cut this torus at the edges a and b, as depicted in the Figure. We flatten the resulting surface on a plane, triangulate and label it as shown in the Figure. The resulting triangulation is a valid Δ-complex. It is made of one 0-simplex v, three 1-simplices a, b and c and two 2-simplices U and L. The arrows on the simplices indicate the

orientations on the simplices. Finally, note that it is possible to assemble a torus from this simplicial complex by the identification of the multiple edges a and b, centered at v.

Let us start with the zero-th homology group. With only one vertex v, $C_0(T) \simeq \mathbb{Z}$. Similarly, $C_1(T) \simeq \mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z}$, indicating the free group on the three edges a, b, c. Any $c \in C_1(T)$ can be expressed as $c = \alpha a + \beta b + \gamma c$, where $\alpha, \beta, \gamma \in \mathbb{Z}$. Clearly, the boundary map $\partial_1 : C_1(T) \to C_0(T)$ is zero. To see this, note that $\partial_1(c) = \alpha\partial_1(a) + \beta\partial_1(b) + \gamma\partial_1(c) = \alpha(v - v) + \beta(v - v) + \gamma(v - v) = 0$. Therefore,

$$H_0^{\Delta}(T) \simeq \text{ker}\partial_0/\text{im}\partial_1 \simeq C_0(T) \simeq \mathbb{Z}.$$

This is consistent with the observation that the space has only connected component.

Consider another basis for $C_1(T)$ as $\{a, b, a + b - c\}$. Since $\partial_2(U) = \partial_2(L) = a + b - c$, it follows that

$$H_1^{\Delta}(T) \simeq \text{ker}\partial_1/\text{im}\partial_2 \simeq \mathbb{Z} \oplus \mathbb{Z},$$

by modding out the component in the free group $C_1(T)$, corresponding to $a + b - c$. This leaves a and b as the representative cycles for the two non-trivial homology classes in the first homology group.

Since there are no simplices for dimension 3 or higher, $C_k(T) \simeq 0$ for $k > 2$. From this it follows that

$$H_2^{\Delta}(T) \simeq \text{ker}\partial_2/\text{im}\partial_3 \simeq \text{ker}\partial_2 \simeq \mathbb{Z}.$$

This is the free group generated by $U - L$, since for any $\alpha U + \beta L \in C_2(T)$, $\partial_2(\alpha U + \beta L) = (\alpha + \beta)(a + b - c) = 0$ if and only if $\alpha = -\beta$. Finally, $H_k^{\Delta}(T) \simeq 0$ for $k > 2$. To summarize,

$$H_n^{\Delta}(T) \simeq \begin{cases} \mathbb{Z} \oplus \mathbb{Z}, & \text{for } n = 1; \\ \mathbb{Z}, & \text{for } n = 0.2; \\ 0, & \text{for } n \geq 0. \end{cases}$$

Note that each Δ-complex can be transformed into a simplicial complex (the likes of which we have encountered in this work). This can be done using a technique called as barycentric subdivision. It can be shown that the second barycentric subdivision of any Δ-complex produces a simplicial complex, which is homeomorphic to the original Δ-complex.

Without going into details, it is enough to understand that the simplicial complexes are Δ-complexes whose simplices are uniquely determined by their vertices. In Δ-complexes, this restriction is not in force. This is the reason that the Δ-complex representation of the torus drawn in Figure 57 has only two 2-simplices. A simplicial complex representation of the torus, however, would require at least 14 triangles, 21 edges and 7 vertices. The Δ-complex representation, therefore makes the computations much easier in many cases.

A.5 Singular Homology and Homotopy Invariance

Simplicial homology is a very powerful theory. However, there is a more elegant homology theory, known as singular homology, which lets us study many questions in a more straightforward manner. Many of the results developed in singular homology also carry for Δ-complexes and in many cases for simplicial complexes as well. We introduce this theory below.

A *singular n-simplex* in a space X is a continuous map (instead of a set) given by $\sigma : \Delta^n \to X$. With the set of all such singular n-simplices as a basis, one can generate a free Abelian group $C_n(X)$, whose elements are called the *chains*. Each n-chain can be written as a finite formal sum $\sum_i n_i\sigma_i$ for $n_i \in \mathbb{Z}$ and $\sigma_i : \Delta^n \to X$. Similarly the boundary map between singular chains $\partial_n : C_n(X) \to C_n(X)$ is given by

$$\partial_n(\sigma) = \sum_i (-1)^i \sigma|_{[v_0,\cdots,\hat{v_i},\cdots,v_n]}.$$

A similar proof to the one presented for simplicial homology shows that $\partial_n\partial_{n+1} = 0$. Therefore, the singular homology groups are given by

$$H_n(X) = \ker\, \partial_n/\text{im}\, \partial_{n+1}.$$

On the face of it, the singular homology theory looks very similar to simplicial homology. However, there are many subtle differences. We briefly summarize some results from singular homology theory which may have some analogs in simplicial homology, but are easier to derive in the singular theory.

If of a space X has path-wise connected components X_α, then

$$H_n(X) \simeq \oplus_\alpha H_n(X_\alpha).$$

If X is path-wise connected, then $H_0(X) \simeq \mathbb{Z}$. For a space with multiple components, $H_0(X)$ is a direct sum of $\mathbb{Z}$'s for each component of X. If X is homotopic to a point, then $H_n(X) = 0$ for $n > 0$. For detailed proofs, please see [49].

We now present a result which is particularly important for this work: Spaces that are homotopy equivalent have isomorphic homology groups.

Corresponding to each map between spaces $f : X \to Y$, there is an induced homomorphism between their respective chain groups denoted by $f_\sharp : H_n(X) \to H_n(Y)$ for each n. This can be defined in the following way. Since each singular n-simplex is given by $\sigma : \Delta^n \to X$, we compose it with f to get

$$f_\sharp(\sigma) = f\sigma : \Delta^n \to Y.$$

This can extended linearly over any chain in $C_n(X)$ to get

$$f_\sharp(\sum_i n_i\sigma_i) = \sum_i n_i f_\sharp(\sigma_i) = \sum_i n_i f\sigma_i.$$

Let us now see how this maps behaves with the boundary operators.

$$\begin{aligned} f_\sharp\partial(\sigma) &= f_\sharp\left(\sum_i (-1)^i \sigma|_{[v_0,\cdots,\hat{v_i},\cdots,v_n]}\right), \\ &= \sum_i (-1)^i f\sigma|_{[v_0,\cdots,\hat{v_i},\cdots,v_n]} \\ &= \partial f_\sharp(\sigma). \end{aligned}$$

This means that $f_\sharp\partial = \partial f_\sharp$. Therefore, we have the following commutative diagram:

$$\begin{array}{ccccccccc} \cdots & \to & C_{n+1}(X) & \xrightarrow{\partial} & C_n(X) & \xrightarrow{\partial} & C_{n+1}(X) & \to \cdots \\ & & \downarrow f_\sharp & & \downarrow f_\sharp & & \downarrow f_\sharp & \\ \cdots & \to & C_{n+1}(Y) & \xrightarrow{\partial} & C_n(Y) & \xrightarrow{\partial} & C_{n+1}(Y) & \to \cdots \end{array}$$

Consider a cycle α, i.e. $\partial\alpha = 0$. Then

$$\partial(f_\sharp\alpha) = f_\sharp(\partial\alpha) = 0.$$

In other words, $f_\sharp$ takes cycles in X to cycles in Y. Also, if $\partial\beta$ is a boundary in X,

$$f_\sharp(\partial\beta) = \partial(f_\sharp\beta),$$

which is boundary in Y. This proves that $f_\sharp$ is a *chain map*, i.e. it induces a homomorphism between the respective homology groups

$$f_* : H_n(X) \to H_n(Y),$$

which satisfies two elementary properties

1. The identity map $\mathrm{id} : X \to X$ induces the identity map id_* on the homology groups.
2. The composition of two maps $X \xrightarrow{g} Y \xrightarrow{f} Z$ induces the composition of the induced homomorphisms: $(gf)_* = g_* f_*$.

Finally, we give the following result.

Theorem A.5.1 *If two maps $f, g : X \to Y$ are homotopic, then they induce the same homomorphism $f_* = g_* : H_n(X) \to H_n(Y)$.*

For a detailed proof of this theorem, we refer the reader to [49]. The main ingredient of the proof is a method of subdividing $\Delta^n \times [0,1]$ into $n+1$ simplices and the use of a certain prism operator P as a chain homotopy between $g_\sharp$ and $f_\sharp$. First, the following relation is derived.

$$\partial P = g_\sharp - f_\sharp - P\partial.$$

Then, consider a cycle $\alpha \in C_n(X)$. Since $\partial\alpha = 0$, we have

$$g_\sharp(\alpha) - f_\sharp(\alpha) = \partial P(\alpha) + P\partial(alpha) = \partial P(\alpha).$$

This means that $g_\sharp(\alpha) - f_\sharp(\alpha)$ is a boundary, which means that both $g_\sharp(\alpha)$ and $f_\sharp(\alpha)$ define the same homology class. Therefore $f_*(\alpha) = g_*(\alpha)$, proving the theorem.

From these properties of f_*, g_* we immediately get our main result.

Corollary A.5.1 *The maps $f_* : H_n(X) \to H_n(Y)$ induced by a homotopy equivalence $f : X \to Y$ are isomorphisms for all n.*

A.6 Relative Homology Groups and Exact Sequences

Relative homology groups are useful tools for studying quotient spaces. Let A be a subspace of a space X and denote by $C_n(X, A)$ the quotient chain group $C_n(X)/C_n(A)$. Therefore any chain inside A is considered to be trivial in $C_n(X, A)$. The boundary map $\partial_n : C_n(X) \to C_{n-1}(X)$ induces a quotient boundary map $\partial_n : C_n(X, A) \to C_{n-1}(X, A)$. Here too, $\partial_n \partial_{n+1} = 0$ holds. Therefore one can define the *relative homology groups*, $H_n(X, A) = \ker \partial_n / \text{im } \partial_{n+1}$ using these boundary operators in exactly the same manner. It should be noted that

1. Elements of $H_n(X, A)$ are called *relative cycles*. They are n-chains $\xi \in C_n(X)$ such that $\partial\xi \in C_{n-1}(A)$.

2. A cycle α in called *trivial*, if it is a *relative boundary*. In other words, $\alpha = \partial\beta + \gamma$ for some $\beta \in C_{n+1}(X)$ and $\gamma \in C_n(A)$.

It can be shown that these chain groups satisfy the following commutative diagram.

$$\begin{array}{ccccccccc} 0 & \to & C_n(A) & \xrightarrow{i} & C_n(X) & \xrightarrow{j} & C_n(X, A) & \to 0 \\ & & \downarrow \partial & & \downarrow \partial & & \downarrow \partial & \\ 0 & \to & C_{n-1}(A) & \xrightarrow{i} & C_{n-1}(X) & \xrightarrow{j} & C_{n-1}(X, A) & \to 0 \end{array}$$

where i is the inclusion map and j is a quotient map with respect to A. From this, it can be shown that relative homology groups $H_n(X, A)$ for any pair $(X, A \subset X)$ satisfy the *long exact sequence*

$$\cdots \to H_n(A) \xrightarrow{i_*} H_n(X) \xrightarrow{j_*} H_n(X, A) \xrightarrow{\partial} H_{n-1}(A) \xrightarrow{i_*} H_{n-1}(X) \to \cdots \to H_0(X, A) \to 0$$

(Recall the definition of exactness from the first section on Abelian groups). Notice, that this sequence is defined for *any* pair (X, A). One might wonder, as to why not define the homology groups for the quotient space X/A directly. It can be shown that we do have a long exact sequence,

$$\cdots \to H_n(A) \xrightarrow{i_*} H_n(X) \xrightarrow{j_*} H_n(X/A) \xrightarrow{\partial} H_{n-1}(A) \xrightarrow{i_*} H_{n-1}(X) \to \cdots$$

However, the existence of such a sequence requires that (X, A) is a *good pair*, namely that A is a non-empty closed subspace that is a deformation retract of some neighborhood in X. The long exact sequence for the relative homology groups, however, holds for any pair, and is therefore preferred over the the exact sequence for the homology of the quotient space X/A.

Finally, it is appropriate to mention the equivalence of simplicial and singular homology for a Δ-complex X. One can define a homomorphism $\theta : \Delta_n(X) \to C_n(X)$ between the two chain groups by sending each n-simplex of X to its characteristic map $\sigma : \Delta^n \to X$. From this one can get a canonical homomorphism between the respective homology groups. One can prove the following general result.

Theorem A.6.1 *The induced homomorphisms, $H_n^{\Delta}(X, A) \to H_n(X, A)$, are isomorphisms for all n and all Δ-complex pairs (X, A).*

For a detailed proof we refer the reader to [49].

REFERENCES

[1] A. Abrams and R. Ghrist, "Finding Topology in a Factory: Configuration Spaces," *American Mathematical Monthly*, Vol. 109, No. 2 , pp. 140–150, 2002.

[2] J. Agre and L. Clare, "An Integrated Architecture for Cooperative Sensing and Networks," *Computer*, Vol. 33, pp. 106–108, May 2000.

[3] A. Ames and S. Sastry, "A Homology Theory for Hybrid Systems: Hybrid Homology," in *Proc. of the 8th International Workshop on Hybrid Systems: Computation and Control*, Zurich, Switzerland, March 9-11, 2005.

[4] E. Arkin, M. Held, J. Mitchell, and S. Skiena, "Hamiltonian Triangulations for Fast Rendering," *Proc. Algorithms-ESA'94*, LNCS 855, pp. 36–47, September 1994.

[5] M. Armstrong, *Basic Topology*, Springer-Verlag, 1983.

[6] H. Atlan, "Measures of Complexity," *Lecture Notes in Physics*, vol 314, pp 112–12, 1988.

[7] H. Axelsson, A. Muhammad, and M. Egerstedt, "Autonomous Formation Switching for Multiple, Mobile Robots," in *Proc. IFAC Conference on Analysis and Design of Hybrid Systems,* Sant-Malo, Brittany, France, June 2003.

[8] T. Balch and R.C Arkin, "Behavior-based Formation Control for Multirobot Teams," *IEEE Transaction on Robotics and Automation*, Vol. 14, pp. 926–939, 1998.

[9] Y. Bar-Yam, *Dynamics of Complex Systems (Studies in Nonlinearity)*, Westview Press, 1997.

[10] S. Basu, R. Pollack and M. Roy, *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics Series, Vol. 10, Springer, 2003.

[11] D. Bean, "Recursive Eulerian and Hamiltonian Paths," in *Proc. American Mathematical Society*, Vol. 55, pp. 285–294, 1976.

[12] R. W. Beard, J. Lawton and F. Y. Hadaegh, "A Coordination Architecture for Spacecraft Formation Control," *IEEE Transactions on Control Systems Technology*, Vol. 9, pp. 777–790, 2001.

[13] M. Bern et al., "Emerging Challenges in Computational Topology," Report from the NSF-funded Workshop on Computational Topology, Miami Beach, FL, June 11-12, 1999.

[14] S. H. Bertz, "Complexity: Introduction and Fundamentals," *Mathematical Chemistry*, vol. 7, pp 91–156, 2003.

[15] J. Bochnak, M. Coste, M. Roy, *Real Algebraic Geometry,* Springer-Verlag, Berlin, 1998.

[16] D. Bonchev *Information Theoretic Indices for Characterization of Chemical Structures*, Research Studies Press, 1983.

[17] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Martix Inequalities in Systems and Control Theory*, SIAM Studies in Applied Mathematics, 1994.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambdrige University Press, 2004.

[19] L. Beinkeke and R. Wilson, *Graph Connections: Relationship between Graph theory and other areas of Mathematics*, Oxford, 1997.

[20] J. Bochnak, M. Coste, and M-F. Roy, *Real Algebraic Geometry*, Springer, 1998.

[21] R. Bott and L. Tu, *Differential Forms in Algebraic Topology* Springer-Verlag, Berlin, 1982.

[22] R. Brockett, "Dynamical Systems and Their Associated Automata," *Systems and Networks: Mathematical Theory and Applications*, Vol. 77, pp. 49–69, Akademi-Verlag, Berlin, 1994.

[23] Computational Homology Program (CHomP). Home page, `http://www.math.gatech.edu/~chom/`, 2004.

[24] M. Cover and J. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, 1991.

[25] V. de Silva, R. Ghrist and A. Muhammad, "Blind Swarms for Coverage in 2-D," *Robotics: Science and Systems*, MIT, Cambridge, MA, 2005.

[26] V. de Silva and R. Ghrist, "Coordinate-Free Coverage in Sensor Networks with Controlled Boundaries via Homology." (In preparation)

[27] V. de Silva and R. Ghrist, "Coverage in Sensor Networks via Persistent Homology." (In preparation)

[28] C. Detrain, J. L. Denebourg, and J. M. Pasteels, *Information Processing in Social Insects*, Birkhauser, 1999.

[29] R. Diestel, *Graph Theory*, vol. 173 of Graduate Texts in Mathematics. Springer-Verlag, 2000.

[30] E. Dijkstra, "A note on two problems in connexion with graphs," in *Numerische Mathematik*, vol. 1, pp. 269271, 1959.

[31] U. Dogrusoz and M.S. Krishnamoorthy, "Hamiltonian Cycle Problem for Triangle Graphs," *Technical Report*, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy, NY , USA, 1995.

[32] M. Egerstedt., A. Muhammad and X. Hu, "Formation Control Under Limited Sensory Range Constraints," in *Proc. of the 10th Mediterranean Conference on Control and Automation*, Lisabon, Portugal, 2002.

[33] M. Egerstedt and R.W. Brockett, "Feedback Can Reduce the Specification Complexity of Motor Programs," *IEEE Transactions on Automatic Control*, Vol. 48, No. 2, pp. 213–223, 2003.

[34] M. Egerstedt and X. Hu, "Formation Constrained Multi-Agent Control," *IEEE Transactions on Robotics and Automation*, Vol. 17, pp. 947–951, 2001.

[35] T. Eren, P. Belhumeur and A. Morse, "Closing Ranks in Vehicle Formations Based on Rigidity," in *Proc. of the 41st IEEE Conference on Decision and Control*, Vol. 3, pp. 2959–2964, 2002.

[36] M. Farber, "Topological Complexity of Motion Planning," *Discrete Computational Geometry*, Vol. 29 , pp. 211-221, 2003.

[37] M. Farber, S. Tabachnikov, and S. Yuzvinsky, "Topological Robotics: Motion Planning in Projective Spaces," *International Mathematics Research Notices*, Vol. 34, pp. 1853–1870, 2003.

[38] J. Fax and R. Murray, "Graph Laplacians and Vehicle Formation Stabilization," in *Proc. IFAC World Congress*, 2002.

[39] R. Fierro, A. Das, V. Kumar and J. Ostrowski, "Hybrid Control of Formations of Robots," in *Proc. of ICRA*, Vol 1, pp. 157–162, 2001.

[40] v. Gazi and K. Passino, "Stability Analysis of Social Foraging Swarms," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 34, No. 1. pp. 539–557, Feb. 2004.

[41] R. Ghrist, "Configuration Spaces and Braid Groups on Graphs in Robotics, Braids, Links, and Mapping Class Groups," in *Proc. of Joan Birman's 70th Birthday*, AMS/IP Studies in Mathematics, Vol. 24, pp. 29–40, 2001.

[42] R. Ghrist and A. Muhammad, "Coverage and Hole-Detection in Sensor Networks via Homology," *The Fourth International Conference on Information Processing in Sensor Networks* (IPSN'05), UCLA, Los Angeles, CA, 2005.

[43] R. Ghrist, *Private Communication.*

[44] R. Ghrist and D. Koditschek, "Safe Cooperative Robot Dynamics Via Dynamics on Graphs," in *Proc. of the Eigth International Symposium on Robotics Research*, pp. 81–92, 1998.

[45] C. Godsil and G. Royle, *Algebraic Graph Theory*, Springer-Verlag, 2001.

[46] M. Gromov, "Hyperbolic groups," *Essays in Group Theory* (ed. S. Gersten), Mathematical Sciences Research Institute Publications 8 (Springer, New York, 1987) 75–263.

[47] J. Gross J. and T. Tucker, *Topological Graph Theory*, John Wiley, 1987.

[48] J. Harris, *Algebraic Geometry: A First Course*, Springer- Verlag, 1992.

[49] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2002.

[50] D. Hristu-Varsakelis, M. Egerstedt, and P.S. Krishnaprasad, "On The Structural Complexity of the Motion Description Language MDLe," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003.

[51] C. Hsin and M. Liu, "Network Coverage Using Low Duty-Cycled Sensors: Random and Coordinated Sleep Algorithms," in *Proc. International Workshop on Information Processing in Sensor Networks* (IPSN), April 2004.

[52] T. Hungerford, *Algebra,* Springer-Verlag, 1974.

[53] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules," *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp. 988–1001, 2003.

[54] D. Jordan and M. Steiner, "Configuration Spaces of Mechanical Linkages," *Discrete Computational Geometry*, Vol. 22, pp. 297-315, 1999.

[55] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology,* Applied Mathematical Sciences 157, Springer-Verlag, 2004.

[56] M. Kapovich and J. Millson, "Universality Theorems for Configuration Spaces of Planar Linkages," *Topology*, Vol. 41, no. 6, pp. 1051–1107, 2002.

[57] E. Klavins, "Communication Complexity of Multi-Robot Systems," in *Proc. Fifth International Workshop on the Algorithmic Foundations of Robotics*, Nice, France, 2002.

[58] E. Klavins, "Toward the Control of Self-Assemblinng Systems," *Control Problems in Agentics*, pp, 153–168, Springer Verlag, 2002.

[59] E. Klavins, R. Ghrist, and D. Lipsky, "A Grammatical Approach to Self-Organizing Robotic Systems," *IEEE Transactions on Automatic Control*, 2005. (To Appear)

[60] H. Koskinen, "On the Coverage of a Random Sensor Network in a Bounded Domain," in *Proceedings of 16th ITC Specialist Seminar*, pp. 11-18, 2004.

[61] S. Lavalle, *Planning Algorithms*, Cambridge University Press, 2006.

[62] J. Lawton, R. Beard and B. Young, "A Decentralized Approach to Elementary Formation Maneuvers," in *Proc. IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2728–2733, 2000.

[63] J. Leonard and H. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publisher, Boston, MA, 1992.

[64] X. Li, P. Wan, and O. Frieder, "Coverage in Wireless Ad-Hoc Sensor Networks," *IEEE Transaction on Computers*, Vol. 52, No. 6, pp. 753–763, 2003.

[65] Z. Lin, B. Francis, and M. Maggiore, "Necessary and Sufficient Graphical Conditions for Formation Control of Unicycles," *IEEE Transactions on Automatic Control*, Vol. 50, No.1, pp. 121–127, 2005.

[66] B. Liu and D. Towsley, "A Study of the coverage of large-scale sensor networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004.

[67] M. Mataric, M. Nilsson and K. Simsarian, "Cooperative Multi-Robot Box-pushing," in *Proc. of IROS*, Pittsburgh, PA, pp. 556–561, 1995.

[68] The Math- Works Inc., "LMI Control Toolbox," Version 1.0.7, May 2001.

[69] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-Hoc Sensor Network," in *IEEE INFOCOM*, pp. 1380-1387, 2001.

[70] O. Mermoud and M. Steiner, "Configuration Spaces of Weighted Graphs in High Dimensional Euclidean Spaces," *Contributions to Algebra and Geometry*, Vol. 43, No. 1, pp. 27–31, 2002.

[71] M. Mesbahi, "On State-Dependent Dynamic Graphs and Their Controllability Properties," *IEEE Transactions on Automatic Control*, Vol. 50, No. 3, pp. 387 – 392, 2005.

[72] R. Milgram and J. Trinkle, "The Geometry of Configuration Spaces for Closed Chains in Two and Three Dimensions," *Homology, Homotopy and Applications*, Vol. 6, No. 1, pp. 237–267, 2004.

[73] A. Muhammad and M. Egerstedt, "Connectivity Graphs as Models of Local Interactions," in *Proc. of the 43rd IEEE Conference on Decision and Control*, Bahamas, December 2004.

[74] A. Muhammad and Magnus Egerstedt, Decentralized coordination with local interactions: Some new directions, *Cooperative Control*, Springer Lecture Notes in Control and Information Sciences (LNCIS), Vol. 309, 2005.

[75] A. Muhammad and M. Egerstedt, "On the Structure of Connectivity Graphs of Agent Formations," *Technical Report*, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 2003.

[76] A. Muhammad and M. Egerstedt, "On the Structural Complexity of Multi-Agent Agent Formations," in *Proc. American Control Conference*, Boston, Massachusetts, USA, 2004.

[77] A. Muhammad and M. Egerstedt, "Topology and Complexity of Formations," in *Proc. of the 2nd International Workshop on the Mathematics and Algorithms of Social Insects*, Atlanta, Georgia, USA, December 15-17, 2003.

[78] A. Muhammad and M. Egerstedt, "Connectivity Graphs as Models of Local Interactions." *Journal of Applied Mathematics and Computation*, Vol. 168, Issue 1, September 2005, Pages 243-269.

[79] A. Muhammad and Magnus Egerstedt, Positivstellensatz Certificates for Non-Feasibility of Connectivity Graphs in Multi-agent Coordination, 16th IFAC World Congress, Prague, July 4-8, 2005.

[80] A. Muhammad and M. Egerstedt, "Applications of Connectivity Graph Processes in Networked Sensing and Control," Workshop on Networked Embedded Sensing and Control, University of Notre Dame, 2005.

[81] A. Muhammad and M. Egerstedt., Feasibility, Reachability, and Optimal Control of Connectivity Graphs, Submitted to *SIAM Journal on Control and Optimization*.

[82] J. Munkres, *Elements of Algebraic Topology*, Addison Wesley, 1993.

[83] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

[84] H. Ochiai, P. Mitran, H. Poor and V. Tarokh, "Collaborative Beamforming for Distributed Wireless Ad Hoc Sensor Networks," *IEEE Transactions on Signal Processing*. (To appear)

[85] P. Ogren, E. Fiorelli and N. Leonard, "Formations with a Mission: Stable Coordination of Vehicle Group Maneuvers," in *Proc. of the 15th International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[86] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, PhD thesis, California Institute of Technology, May 2000.

[87] P. A. Parrilo and S. Lall, "Semidefinite Programming Relaxations and Algebraic Optimization in Control", *European Journal of Control*, Vol. 9, No. 2, 2003.

[88] M. Randic and D. Plavsic, "On the Concept of Molecular Complexity," *Croatica Chemica ACTA*, Vol. 75, pp. 107–116, 2002.

[89] PLEX : Matlab Library for Simplicial Complexes. Home page `http://math.stanford.edu/comptop/programs/plex/`, 2003.

[90] J. Reif and H. Wang, "Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots," *Robotics and Autonomous Systems*, Vol. 27, No. 3, 1999.

[91] R. Saber and R Murray, "Distributed Cooperative Control of Multiple Vehicle Formations using Structural Potential Functions," *IFAC World Congress*, Barcelona, Spain, 2002.

[92] R. Saber and R. Murray, "Agreement Problems in Networks with Directed Graphs and Switching Topology," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, 2003.

[93] C. E. Shannon, "A Mathematical Theory of Communications," *Bell Systems Technical Journal*, vol. 27, pp 379–423, 1948.

[94] G. Stengle, "A Nullstellensatz and a Positivstellensatz in Semialgebraic Geometry," *Mathematische Annalen*, Vol. 207, pp. 87-97, 1974.

[95] D. Swaroop, J. Hedrick, "String Stability of Interconnected Systems," *IEEE Transactions on Automatic Control* Vol. 41, pp. 349–357, 1996.

[96] H. Tanner, G. J. Pappas, and V. Kumar, "Leader to Formation Stability," *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 3, pp. 443 – 455, June 2004.

[97] H. Tanner, G. Pappas and V. Kumar, "Input-to-State Stability on Formation Graphs,". in *Proc. of the 41st IEEE Conference on Decision and Control*, Vol. 3, pp. 2439–2444, 2002.

[98] H. Tanner, A. Jadbabaie, G. Pappas, "Stable Flocking of Mobile Agents, Part I: Fixed Topology," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, 2003.

[99] H. Tanner, A. Jadbabaie, G. Pappas, "Stable Flocking of Mobile Agents, Part II: Dynamic Topology," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, 2003.

[100] D. Tian, N. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks," in *Proc. of the 1st ACM international workshop on Wireless sensor networks and applications*, Atlanta, Georgia, USA, 2002.

[101] B. Van Veen and K. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE ASSP Magazine*, Vol. 5, Apr. 1988.

[102] L. Vietoris, "Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen," *Math. Ann.* pp. 454-472, vol. 97, 1927.

[103] F. Xue and P. R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks." *Wireless Networks*, pp. 169-181, Vol. 10, No. 2, March 2004.

[104] H. Zhang and J. Hou, "Maintaining Coverage and Connectivity in Large Sensor Networks," *International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless and Peer-to-Peer Networks*, Florida, Feb. 2004.

VITA

Abubakr Muhammad was born in Pakistan, on December 19, 1976. He received his early education at Crescent Model School and Government College Lahore in Pakistan. He then enrolled at the University of Engineering and Technology (UET) Lahore, where he got his BSc in Electrical Engineering in 2000. He worked in the industry till 2001 on various applications of signal processing in radar, sonar and air-traffic control systems. He then moved to Georgia Tech where he received his MS in Electrical Engineering in 2002 and MS in Mathematics in 2005. During his graduate studies, he has also been a visiting research scholar at the Mathematics departments in University of Illinois, Urbana-Champaign and Stanford University. His current research interests concern the application of algebraic topology and differential geometry methods in robotics, sensor networks and networked control systems.